Remarks and Arguments

1. Formality Problems in Office Action.

The Office Action of October 18, 2006 says that it is a response to Applicant's submission of October 12, 2006; the actual date of Applicant's submission is July 24, 2006. The Office Action Summary page at the bottom indicated that a Notice of References Cited was attached, but that document was missing.

2. The Specification is objected to on the basis of Section 132 (a) as alleged new matter.

Specifically the objection is to the addition of material on page 3 in the July 24, 2006 submission, which amendment added language at the end of the first paragraph on page 6 of the original Application. This material has been cancelled; however two minor additions have been made to the text of the original Application, and **this is not new matter as demonstrated below.** Applicant requests reconsideration of this issue.

First, there is abundant support in the original Application for including PID loop controllers in the concepts and equipment originally disclosed:

(1) Just below the middle of page 14:

"The industrial control and monitoring equipment are any make of Single Loop Controller, Programmable Logic Controller (PLC) or Distributed Control System (DCS) from any manufacturer."

(2) Near the bottom of page 14:

"The wizard builder of the present invention can be adapted to work with a very wide range of application programs which control or monitor processes in real time. The wizard builder can produce a setup wizard which can work with a wide range of monitoring and control devices from many different manufacturers and having many different operating characteristics."

(3) The last paragraph on page 15:

"The industrial control and monitoring equipment units could be any piece of industrial equipment which contains an embedded micro-processor and which is able to offload data values that represent the current state of an industrial process. Examples include machine tools, motor drives, robots, intelligent valves and pumps, telemetry outstations, programmable controllers and distributed control systems."

(4) The section on TESTS on pages 16 and 17 of the original Application demonstrates that the Applicant's customer had used the invention with the Siemens PCS 7 system and the Honeywell TDC 3000 system—which persons skilled in the art know to contain a PID loop controller:

"One user of the ExperTune software is employed by Air Products Company as a process engineer and has no knowledge of computer programming. This user created a wizard to connect the ExperTune software to **Honeywell's TDC 3000 distributed control system.** The wizard was to connect to the Honeywell GUS (Global User Station) workstation via the DDE (Dynamic Data Exchange) de-facto interface.

Another user of ExperTune software is employed by Siemens as a process engineer and has no knowledge of computer programming. This user created a wizard to connect the ExperTune software to the **Siemens PCS 7 distributed control system.** The wizard was to connect to the WinCC HCI (Human Computer Interface) program used in PCS 7 via the OPC (OLE for Process Control) de-facto interface.... Thus the wizard builder produces a new and useful result in producing wizards (setup wizards) which can adapt application programs, for example, to particular devices for monitoring or controlling industrial processes in real time, all without requiring computer programming abilities of the user. "

(5) Claim 1 as originally filed on page 19 of the Application contained the following language which a person in the art would readily know to include PID loop controllers:

" ...for adapting an application program to function with devices or sensors monitoring or controlling a process occurring in real-time..."

Secondly, several Exhibits are attached to this RCE with relevant passages marked in the margins: Exhibit A summarizing how material on websites **support the idea that PID loop controllers are "controllers commonly used"**, and Exhibits B-P which support

this idea. Specifically, **Exhibit D shows that a PID loop controller is contained in the Siemens PCS 7 distributed control system** which was in the original Application and is mentioned in section (4) above. **Exhibits L through P all clearly show that the Honeywell TDC 3000 Distributed Control System referred to above and referred to in the Application as filed contains PID loop controllers. In fact, one of these Exhibits (Exhibit L) is *U. S. Patent 5,697,436* !**

<u>There is abundant support in the patent law and the MPEP for allowing an Applicant to add terms to the Specification in connection with new terms introduced in claims which have been narrowed, if these terms are well known to those skilled in the art if they are inherent in the disclosure of the original application.</u>

MPEP Section 608.04(a) states that matter not in the original specification, claims or drawings **<u>may</u>** be new matter depending on the circumstances. The addition of inherent characteristics such as physical properties *may* be new matter. MPEP Section 2163 deals with the written description requirement and the new matter requirement together since they are closely related. MPEP Section 2163 I second paragraph states:

> "Much of the written description case law addresses whether the specification as originally filed supports claims not originally in the application. The issue raised in the cases is most often phrased as whether the original application provides "adequate support" for the claims at issue or whether the material added to the specification incorporates "new matter" in violation of 35 U.S.C. 132."

The following paragraph of the MPEP states:

> "**Possession** may be shown in a variety of ways including description of an **actual reduction to practice...**"

In the present Application, the TESTS Section on pages 16 and 17 of the original Application showed that the invention of the Application referred to the Siemens PCS 7 distributed control system, and the Honeywell TDC 3000 system, which contain PID controllers as demonstrated above. Therefore the Applicant **was in possession** of the Invention as it related to PID loop controllers, and it worked with such controllers.

MPEP Section 2163 I B second paragraph states:

> While there is **no *in haec verba* requirement**, newly added claim limitations must be supported in the specification through express, **implicit, or inherent** disclosure.

In other words, the original disclosure **need not recite all claim terms in the same exact words, if those terms are inherent** in the original disclosure. Applicant strongly asserts that a person skilled in the art would readily recognize that the mention of the Siemens and Honeywell systems as discussed above contained inherent disclosure of PID loop controllers because those systems are widely known to contain such controllers.

MPEP Section 2163 II A 3(a) states:

> **What is conventional or well known to one of ordinary skill in the art need not be disclosed in detail.** See *Hybritech Inc. v. Monoclonal Antibodies, Inc.*, 802 F.2d at 1384, 231 USPQ at 94. >See also *Capon v. Eshhar*, 418 F.3d 1349,

1357, 76 USPQ2d 1078, 1085 (Fed. Cir. 2005)("The 'written description' requirement must be applied in the context of the particular invention and the state of the knowledge.... As each field evolves, the balance also evolves between what is known and what is added by each inventive contribution."). **If a skilled artisan would have understood the inventor to be in possession of the claimed invention at the time of filing, even if every nuance of the claims is not explicitly described in the specification, then the adequate description requirement is met.** See, e.g., *Vas-Cath*, 935 F.2d at 1563, 19 USPQ2d at 1116; *Martin v. Johnson*, 454 F.2d 746, 751, 172 USPQ 391, 395 (CCPA 1972) (stating **"the description need not be in *ipsis verbis* [i.e., "in the same words"] to be sufficient").....**

The description needed to satisfy the requirements of 35 U.S.C. 112 **"varies with the nature and scope of the invention at issue, and with the scientific and technologic knowledge already in existence."** *Capon v. Eshhar*, 418 F.3d at 1357, 76 USPQ2d at 1084.

MPEP Section 2163 II A 3(b) states regarding new or amended claims:

The examiner has the initial burden of **presenting evidence or reasoning to explain why persons skilled in the art would not recognize** in the original disclosure a description of the invention defined by the claims....

To comply with the written description requirement of 35 U.S.C. 112, para. 1, or to be entitled to an earlier priority date or filing date under 35 U.S.C. 119, 120, or 365(c), each claim limitation must be expressly, **implicitly, or inherently** supported in the originally filed disclosure. When an explicit limitation in a claim

"is not present in the written description whose benefit is sought it must be **shown that a person of ordinary skill would have understood, at the time the patent application was filed, that the description requires that limitation.**" *Hyatt v. Boone*, 146 F.3d 1348, 1353, 47 USPQ2d 1128, 1131 (Fed. Cir. 1998).

The Examiner has **<u>not made a prima facie case</u>** that persons skilled in the art would not recognize the original application as inherently disclosing PID controllers. The **numerous** Exhibits above, which are **posted on the Internet**, conclusively show at that the information about the Siemens and Honeywell systems **<u>inherently discloses</u>** PID loop controllers to those of ordinary skill in the art.

Finally, MPEP Section 2163.07(a) clearly supports the above discussion that when the original application **<u>inherently discloses</u>** an idea such as PID loop controllers, the **<u>application may be later amended to recite the idea without introducing new matter:</u>**

2163.07(a) Inherent Function, Theory, or Advantage

By disclosing in a patent application a device that **inherently performs a function or has a property**, operates according to a theory or has an advantage, a patent application necessarily discloses that function, theory or advantage, **even though it says nothing explicit concerning it. The application may later be amended to recite the function, theory or advantage without introducing prohibited new matter.** *In re Reynolds*, 443 F.2d 384, 170 USPQ 94 (CCPA 1971); *In re Smythe*, 480 F. 2d 1376, 178 USPQ 279 (CCPA 1973). "To establish

inherency, the extrinsic evidence 'must make clear that the missing descriptive matter is necessarily present in the thing described in the reference, and that it would be so recognized by persons of ordinary skill.

Therefore, the minor additions to the Specification referring to PID loop controllers are appropriate, and do not constitute addition of prohibited new matter. Consequently, the claims now have support in the Specification, and the 112 issue is removed.

3. Claims 1-12, and 14-20 are rejected under Section 112 first paragraph for failing the written description requirement and for failure to show that the Applicant had possession of the claimed invention when the Application was filed

As discussed above, it is entirely appropriate, **and not new matter**, to add the above brief references in the Specification to PID loop controllers. The Specification now **contains support** for the PID loop controllers in claim 1. This is fully supported by the MPEP and other citations above. The fact that the Applicant **had possession** of the invention is shown by the TESTS Section of the original application which referred to the Siemens PCS 7 distributed control system, and the Honeywell TDC 3000 system, which contain PID controllers.

4. Claims 1-7, 9-12, 14, and 16-20 are rejected under 103 on the basis of Gillis, and Thomas.

The claims have not been amended since the July 24 submission in which the claims were narrowed to focus on PID loop controllers. There is support in the original Application in the first paragraph on page 6 which refers to "controllers which are commonly used", and in this RCE it is demonstrated above that PID loop controllers are well know to be included in that term. The Specification has been amended above to make this concept more explicit. Applicant believes that this **term "PID loop controller", or any equivalent concept, is not found in any of the patents** cited against the claims. Therefore, **all elements and limitations of the claims** rejected under 102 and 103 are **not found** in the cited patents, or in the combinations of patents. Therefore, the rejection of the claims under both 102 and 103 should be withdrawn. The Office Action of October 18, 2006 simply repeated apparently word for word the previous claim rejections and did not take into account that the claims have been narrowed to focus on PID loop controllers. Since adequate support has been added to the Specification for PID loop controllers, these rejections should be reviewed, and since none of the cited patents mention a PID loop controller, or equivalent, at any point, these rejections should be withdrawn.

Notwithstanding the above, the Applicant does not understand the sentence spanning pages 4 and 5 of the Office Action about motivation. Applicant disagrees that there would be such a motivation since Thomas is from a **radically different field**—electrical distribution networks, not control of physical industrial processes. Such a motivation to combine the patents would **not have been obvious** to a person of ordinary skill in the art

at the time of the invention. There is **no mention anywhere in Thomas of PID** loop controllers since these are not adapted to controlling electrical distribution systems.

5. Claims 8 and 15 are rejected under 103 on the basis of Gillis, Thomas and Gauthier.

As discussed above, the claims have not been amended since the July 24 submission in which the claims were narrowed to focus on PID loop controllers. There is support in the original Application in the first paragraph on page 6 which refers to "controllers which are commonly used", and in this RCE it is demonstrated above that PID loop controllers are well know to be included in that term. The Specification has been amended above to make this concept more explicit. Applicant believes that this term **"PID loop controller", or any equivalent concept, is not found in any of the patents cited** against the claims. Therefore, **all elements and limitations of the claims** rejected under 102 and 103 are **not found** in the cited patents, or in the combinations of patents. Therefore, the rejection of the claims under both 102 and 103 should be withdrawn. The Office Action of October 18, 2006 simply repeated apparently word for word the previous claim rejections and did not take into account that the claims have been narrowed to focus on PID loop controllers. Since adequate support has been added to the Specification for PID loop controllers, these rejections should be reviewed, and since none of the cited patents mention a PID loop controller, or equivalent, at any point, these rejections should be withdrawn.

6. Failure to show motivation to combine references regarding rejections in Section 4 and 5 above.

A major new case from the U. S. Court of Appeals for the Federal Circuit (In Re Kahn, 04-1616 (Serial No. 08/773,282, Decided March 22, 2006) has emphasized that to support an obviousness rejection, the Examiner **cannot make broad conclusory statements** that the invention is obvious from cited patents, but **must give reasons** for that statement, and **must give reasons for the existence of a motivation to combine**. The following excerpts from the case contain these points:

> "Most inventions arise from a combination of old elements and each element may often be found in the prior art. Id. at 1357. However, mere identification in the prior art of each element is insufficient to defeat the patentability of the combined subject matter as a whole. Id. at 1355, 1357. Rather, to establish a prima facie case of obviousness based on a combination of elements disclosed in the prior art, the Board must articulate the basis on which it concludes that it would have been obvious to make the claimed invention. Id. In practice, this requires that the Board "explain the reasons one of ordinary skill in the art would have been motivated to select the references and to combine them to render the claimed invention obvious." Id. at 1357-59. This entails consideration of both the "scope and content of the prior art" and "level of ordinary skill in the pertinent art" aspects of the Graham test.
>
> When the Board does not explain the motivation, or the suggestion or teaching, that would have led the skilled artisan at the time of the invention to the claimed combination as a whole, we infer that the Board used hindsight to conclude

that the invention was obvious. Id. at 1358. The "motivation-suggestion-teaching" requirement protects against the entry of hindsight into the obviousness analysis, a problem which § 103 was meant to confront. ... By requiring the Board to explain the motivation, suggestion, or teaching as part of its prima facie case, the law guards against hindsight in all cases—whether or not the applicant offers evidence on secondary considerations—which advances Congress's goal of creating a more practical, uniform, and definite test for patentability. ... The motivation-suggestion-teaching test picks up where the analogous art test leaves off and informs the Graham analysis. To reach a non-hindsight driven conclusion as to whether a person having ordinary skill in the art at the time of the invention would have viewed the subject matter as a whole to have been obvious in view of multiple references, the Board must provide some rationale, articulation, or reasoned basis to explain why the conclusion of obviousness is correct. The requirement of such an explanation is consistent with governing obviousness law, see § 103(a); Graham, 383 U.S. at 35; Dann, 425 U.S. at 227-29, and helps ensure predictable patentability determinations.... However, rejections on obviousness grounds cannot be sustained by mere conclusory statements; instead, there must be some articulated reasoning with some rational underpinning to support the legal conclusion of obviousness. ... Therefore, the "motivation-suggestion-teaching" test asks not merely what the references disclose, but whether a person of ordinary skill in the art, possessed with the understandings and knowledge reflected in the prior art, and motivated by the general problem facing the inventor, would have been led to make the combination recited in the claims.

On page 9 of the Office Action it is stated that the motivation to combine Thomas with the other patents is disclosed in paragraph 0004 of Thomas. The Office Action gives **no explanation of this statement whatsoever**. Applicant strongly disagrees since no reasoning has been given, and Thomas is from a **radically different field**, namely controlling electrical distribution systems. PID loop controllers are not adapted to that field. In addition, the cited cases (*Fine* and *Jones*) are **not relevant** to the present Application because they **held against a conclusion of obviousness.** Therefore, if the Examiner repeats any 103 rejections based on an alleged motivation to combine, Applicant **requests that the required reasons in light of the Kahn case be given**.

Summary

1. Brief amendments to the Specification incorporating PID loop controllers are permitted by the MPEP and other authorities since they were **inherently** disclosed in the original Application by references to Siemens and Honeywell systems, and other references.

2. These additions are not new matter under the MPEP.

3. The existing claims now have support for the term PID loop controller, and the Section 112 issue is removed.

4. Rejections of the claims under Section 103 should be withdrawn because none of the patents refers to PID loop controllers or any equivalent.

5. Notwithstanding 4 above, no adequate reasoning or explanation was given about a motivation to combine the patents. Thomas is from a radically different field.

6. Possession of the invention by Applicant is shown by reduction to practice in the TESTS section of the original Application.

Conclusion

Because of the above amendments, and because of the above discussion and arguments, Applicant respectfully submits that the Application, with claims as amended above, is now in condition for allowance, and that action is urgently requested. Enclosed is a check for the fee required by the attached Petition for Extension of Time.

Respectfully Submitted,

Henry L. Smith, Jr.

Henry L. Smith, Jr.

Reg. # 31,425

FEB. 16, 2007

9273 Cornell Circle

Highlands Ranch, CO 80130-4141

Telephone (303) 346-5045

Fax (303) 346-5045

OIPE IAP11
FEB 2 0 2007
PATENT & TRADEMARK OFFICE

EXPRESS MAIL CERTIFICATE

Express Mail Label # ED448576393US

Date of Deposit: 2007, FEB. 16

I hereby certify that this paper or fee is being deposited in the United States Postal Service using "Express Mail Post Office to Addressee" service under 37 CFR 1.10 on the date indicated above and is addressed to Mail Stop RCE, Commissioner for Patents, P. O. Box 1450, Alexandria, VA 22313-1450.

Signed: Henry L. Smith, Jr.

Name: Henry L. Smith, Jr., #31,425 Date: FEB. 16, 2007

EXHIBIT A

APPLICATION 10/034467

The idea is to prove that PID Controllers are clearly included in concepts and information in the Application as originally filed, and those skilled in the art would recognize that.

The PID controllers are a subset of "devices or sensors monitoring or controlling a process occurring in real-time." (See page 19 Claim 1 of the Application as filed.)

See section on Tests on pages 16-17 of the original Application which mentions the Siemens PCS 7 distributed control system.

****

A page from an acknowledged PID control expert.
"PID is the control algorithm most often used in industrial control.
It is implemented in industrial single loop controllers, distributed control systems (DCS) and programmable logic controllers (PLC)."

http://www.learncontrol.com/pid/

****

Confirmation that PID controllers are widely used in process industries.
"at this very moment some 95 % of all controllers in the process industry are still of the PID-type."

http://www.ipcos.be/products/generic/rapid.html

****

Confirmation that PID Controllers are a constituent part of
"Siemens
PCS 7 distributed control system" (Application page 17)
"Modifications can now be propagated throughout the system
faster
and while online with PCS 7. Adding a new process object,
such
as a
PID control loop, takes only a few minutes. "

http://www2.sea.siemens.com/Products/Process-
Automation/Product/SIMATIC+
PCS+7/Reduced+Engineering.htm

****

Advert for a training course about PID controllers.
Diagram of
how
a PID controller operates.

http://www.bin95.com/PID_Process_Control_Saint-Louis.htm

****

Confirmation that PID controllers are a constituent part of
a
Siemens Simatic S7 PLC.
"Example for using FB41 PID-Control for the SIMATIC S7
PLC's
S7-300/400. "

http://www1.control.com/PLCArchive

****

More confirmation that PID contrllers are a constituent
part of
Single Loop Controllers and Programmable Logic Controllers
(PLCs)
"Which one is better to design a process plant which has
several
PID
controlling loops:
1) Connect all loops to the plant PLC, do some PID
programming,
and
let the PLC control the loops directly.
2) Utilise single-loop controllers (with PID capability)
for
every
single loop,..."

http://www.control.com/thread/1026228393

****

More confirmation that PID control is a very common
constituent
of
PLCs and has been for many years.
"PLC Programming
South Coast Systems has 25 years experience in PLC code
development
specializing in Allen-Bradley, GE, Modicon, and TI/Siemens
PLC
systems. ....
With the standards SCS has developed over the years for
motor,
valve, and PID control,....."

http://www.southcoastsystems.com/plc.html

****

Extract from wikipedia entry on Programmable Logic
Controllers.

"PLCs may include logic for single-variable feedback analog control
loop, a "proportional, integral, derivative" or "PID controller." A
PID loop could be used to control the temperature of a manufacturing
process, for example. Historically PLCs were usually configured
with
only a few analog control loops; where processes required hundreds
or thousands of loops, a distributed control system (DCS) would
instead be used. However, as PLCs have become more powerful, the
boundary between DCS and PLC applications has become less clear-cut."

http://en.wikipedia.org/wiki/Programmable_logic_controller

****

Extract from wikipedia entry on Distributed Control Systems.

"A typical DCS consists of functionally and/or geographically
distributed digital controllers capable of executing from 1 to
256
or more regulatory control loops in one control box. The
input/output devices (I/O) can be integral with the controller
or
located remotely via a field network. Today's controllers have
extensive computational capabilities and, in addition to
proportional, integral, and derivative (PID) control, can generally
perform logic and sequential control."

http://en.wikipedia.org/wiki/Distributed_Control_System

****

Extract from wikipedia entry on PID controllers.

"A proportional-integral-derivative controller (PID
controller)
is a
common feedback loop component in industrial control
systems."

http://en.wikipedia.org/wiki/PID_controller

= EXHIBIT B =

Control Information from John Shaw

This site provides information about process control, particularly the PID control algorithm, controller tuning, cascade control, ratio control, and other topics.

PID is the control algorithm most often used in industrial control. It is implemented in industrial single loop controllers, distributed control systems (DCS) and programmable logic controllers (PLC).

We are all entitled to have an opinion. If you don't have one, you can have my opinions for FREE by Clicking Here

Buy and download my eBook: ***The PID Control Algorithm: How It Works, How To Tune It, and How to Use It. 2nd ed.*** 62pp Includes expanded information about cascade, feedforward, and other control schemes. Adobe .pdf file **only $15**	Most major credit cards accepted. ***Get more details, buy, and download***
ControlSim The PID controller and process simulator based on MS Excel spread sheets. **$45.00**	Most major credit cards accepted. ***Get more details, buy, and download***

Questions and answers

B-1

Periodically I publish a question and answer. The questions come from e-mail to john@jashaw.com, are posted on various forums such as control.com or the sci.engr.control news group, or the forum on www.jashaw.com. The answers are those I have provided either on the forum or by e-mail. Often the questions and answers are rewritten to make them more generally applicable. Several questions may be combined into one general question, therefore names will not.usually be used.

- How is the controller action affected if the valve is Air to Close (Fail Open)? NEW!
- Difference between series and parallel derivative and integral
- Derivative on process rather than error
- Proportional vs. Proportional-reset for level control.
- Reset windup and the problems it can cause.
- Best scan time for loops.
- Velocity and Full Value forms of the PID algorithm.
- Interactive and non-interactive integral and derivative.
- Fail safe level switch connections.
- What is the "standard" PID equation? Gain affects all terms?
- What are the benefits of realistic 3-D graphics for operator interface?
- Should derivative be used on temperature loops?

Information about the PID algorithm:
Tutorial on the PID control loop and tuning (in 6 parts)

B-2

This tutorial describes the PID control algorithm, how it works, and how it is tuned.

Brief Explanation of the PID control algorithm

A brief version of the tutorial.

Suggested Code for the PID Algorithm

Two suggestions about how to implement the PID algorithm in digital form.

Information about more complex strategies; Override Control

When there is some other measurement that must be constrained, forcing the output of a controller to follow another controller's output.

Cascade Control

A short explanation, with diagrams, of cascade control and the details of its implementation.

Ratio Control

A short explanation, with diagrams, of ratio control and the details of its implementation.

Control Views

News, articles, and comments from the world of process control.

Working for yourself

Information about consulting, contracting, and self employment.

Know someone interested in PID control? Click here to Email them a link to this site.

Dealing with Uncertainty in Process Design: Monte Carlo Simulation.

B-3

Thousand Acre Swamp Hike Schedule
A schedule of the 2006 public hikes at Thousand Acre Swamp in Penfield, NY

Also of interest:

Control Engineering Information, my new site.

Process Measurement and Control Forum for discussion of all aspects of process measurement and control,
The sci.engr.control news group has discussions of interest to control engineers. Google groups: sci.engr.control
Buy books on control topics. I have a list of books I recommend for reference and for learning about control.

Learn about process control or get involved in the profession. Join ISA

Information about apartments in the Ballantyne area of Charlotte, NC Click Here!

My Personal Web Site - John A. Shaw
My Business Web Site - Process Control Solutions
My e-mail address - john@jashaw.com

B-4

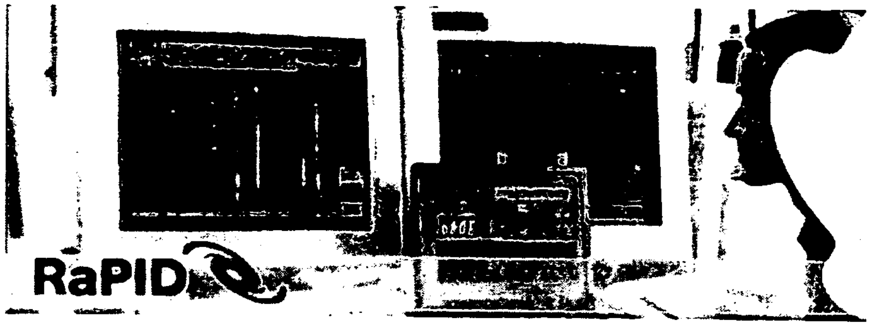
Privacy Policy

Page Updated 05/03/06 19:59

B-5

= EXHIBIT C =

IPCOS *Creators in Control*



→ RaPID overview
→ Benefits
→ RaPID at Work
→ Deliverables
→ Download a full functional demo now (active during 2 weeks).

RaPID, Robust Advanced PID Control

Advanced Process Control or APC is undoubtedly the control philosophy of the future. And yet, at this very moment some 95 % of all controllers in the process industry are still of the PID-type. These PID's will continue to be the workhorse of automation for the coming years. Simply because they are so easy to implement in safety systems such as DCS's and PLC's. This means that, even if you have installed an APC system like INCA, your APC system will continue to act on the plant via the underlying control loops - yes, of the PID kind. *
There is one important lesson to be drawn from this: you need to have your PID loops opti-mally tuned. And this is exactly where RaPID comes in...

The solution for optimal PID control

7 good reasons to fine-tune your PID loops:

1. You increase the stability and the safety of your plant
Your DCS or PLC system has been set up to ensure the stable and secure operation of your plant, via interlocks and stable control loops. And stable operation is what you need, be it at
full or at half load. RaPID offers you stable operation in all operating points.
2. You earn more money
During our own fine-tuning projects we discovered that we could gain up to 1.000.000 euro per year simply by examining the settings of the primary loops.
3. You cut back on downtime
RaPID takes care of the wear of your actuators (pumps, valves, etc.) during the fine-tuning process. This allows you to find a balance between plant performance and actuator wear. And, as you undoubtedly know, less actuator use is tantamount to less downtime
4. You cut down on energy use

C-1

Oscillating processes consume more energy than stable systems, because you fail to keep your process going at its most efficient working point.

5. You get a grip on the consumption of raw material
RaPID guarantees better control - resulting in lesser use of raw material.

6. You can ensure a better quality control
If you have better control of your process you need less blending. On the average 30% of the control loops increase the short-term variability.

7. You reduce manual operation
In many plants about 30% of the control loops are manual. This means that plant operators
have to spend too much time on keeping the plant running. With RaPID, these are bygones, and operators can focus on the optimisation of the plant. A tremendous improvement.

↑ Top

The benefits are obvious

Why use RaPID?
→ RaPID is the first and one of the only tools that optimises PID loops based on engineering specifications. E.g. if your PID controller is responsible for a temperature following a certain profile (tracking), RaPID will optimise the loop for this specific profile tracking. If the PID control loop is responsible for disturbance rejection, the loop will be optimised for that job.
→ RaPID is not restricted to open loop step response tests. Many different test patterns are possible, even closed loop tests. This means that you do not have to take your PID controller in manual to tune the loop.
→ RaPID has a library of PID structures of most frequently used DCS and PLC systems (Siemens, ABB, Honeywell, Emmerson, etc.). The parameters calculated by RaPID can directly be used in the DCS or PLC. Do you happen to work with micro controllers? No problem what-soever. Simply tune your PID controller and RaPID will create a source code, with the para-
meters for your micro controller.
→ RaPID is a generic tool that is not only used in the process industry. Among the users of RaPID you will find not only chemical companies and refineries but also mechatronic com-panies (e.g. high resolution printer driveline control) and universities. Universities can buy RaPID at very interesting rates.
↑ Top

RaPID at work

To ensure an optimally tuned PID loop, you need to carry out the following steps:

1. Collection of process data: RaPID contains a set of modules to connect the most frequently used DCS, PLC and RTDB systems via OPC, DDE and ODBC. RaPID is also able
to retrieve historical data stored in files or databases.
2. Data preprocessing: sometimes it is necessary to preprocess the collected data in order to remove spikes, trends or noise. RaPID contains all the functionalities

C-2

needed to perform these tasks.
3. System identification: in order to obtain good tuning parameters you need to have a dynamic model of the system that has to be controlled. RaPID contains state of the art identification techniques to find the dead times, the order of the system and all dynamic parameters.
4. Control design: in order to obtain optimal PID settings, it is important to define the control tasks of the PID controller. Depending on the complexity of the control algorithm you can tune the PID for disturbance rejection, tracking or both. Once this information is provided, you will be asked to define the platform type (DCS, PLC) and brand the controller
is running on. As soon as a PID algorithm is selected, RaPID will calculate the PID settings needed to solve your problem, depending on your DCS system.
5. Validation: as soon as you've obtained a set of PID parameters from RaPID, just imple-ment them in your DCS and test the loop. RaPID will allow you to compare and analyse the predicted behavior with the real closed loop behavior.

Platform Environment
RaPID runs on PC's with at least a PII 166 MHz processor with Windows 95/98 or NT/2000/XP/2003 platforms, preferably with 128 MB Ram.

↑ Top

Deliverables

General scope of supply:

RaPID is delivered in the form of a perpetual license.
User may tune as much PID loops as desired with one license.

The RaPID license fee includes:
→ RaPID software with file import (Excel, Matlab, Access, TXT, CSV,....)
→ OPC interface
→ RaPID user manual
→ Dongle
→ One year support and upgrades
→ Further support and upgrades beyond one year (optional)

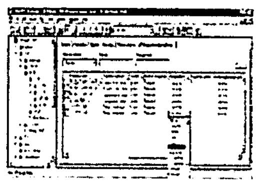
↑ Top

C-3

Printhead

= EXHIBIT D =

Reduced Engineering Costs and Improved Usability



Significant new capabilities have been added to the engineering system to take advantage of SIMATIC® PCS 7's expanded system capacity execution of larger-scale projects.

Concurrent engineering

Multi-project engineering is important for effectively implementing large, complex applications. This feature can be used to break up a large project into smaller project units then be engineered efficiently by several people at the same time (multi-project engineering). The SIMATIC system supports this approach with functions such as Branch & automatically creates text connection placeholders to make it easy to recombine sub-projects. It also provides for administration for sub-projects, which have been locally di central engineering server. This saves valuable time.

Device-oriented configuration

A new tool of the engineering system is the process object view. In combination with the plant hierarchy view, this feature provides a tabular view of all aspects of a process element, such as alarm parameters, I/O signals, operator messages, HMI representation, and archived variables. This tool enables the use of an object-oriented configurati methodology by the process engineer, which is intuitive and easy to use. Configuration time can be reduced by means of bulk engineering functions such as the import/exp objects and creation of templates using standard Microsoft Office tools such as Excel.

This capability enables greater engineering efficiency, helps to avoid configuration errors, and, therefore, enhances productivity in engineering.

D-1

Automatic change management and copying of process areas

With PCS 7, complex process areas, units, or entire production lines can be copied/cloned quickly with minimal retesting required. For instance, a configured and tested uni in its entirety, including all its CFCs, SFCs, pictures, scripts, and archive parameters. This function allows a fully configured and tested unit, consisting of CFCs, SFCs, pictu scripts, archive parameters etc., to be copied or modified by automatically updating the links and connections between each element. It offers enormous cost-saving potenti engineering and for validation, especially for plants with repetitive structures.

Reduced turnaround times

* Modifications can now be propagated throughout the system faster and while online with PCS 7. Adding a new process object, such as a PID control loop, takes only a few minutes. This minimizes waiting time and helps lower the costs of commissioning and operation.

Automatic optimization of control strategy

The system automatically optimizes the execution (scan) sequence of the CFCs, SFCs and FBs contained in the user program. This saves time, improves the effectiveness strategy, and significantly improves resource availability in the controller.

Version cross checker supports plant validation

The version cross checker tool guides the user to quickly distinguish and comprehend the differences between two versions of a project.

D-2

Reduced Engineering Costs - PCS 7 - Microsoft Internet Explorer provided by Comcast

File Edit View Favorites Tools Help

http://www2.sea.siemens.com/Products/Process-Automation/Product/SIMATIC+PCS+7/Reduced+Engineering.htm

Device-oriented configuration

A new tool of the engineering system is the process object view. In combination with the plant hierarchy view, this feature provides a tabular view of all aspects of a process control element, such as alarm parameters, I/O signals, operator messages, HMI representation, and archived variables. This tool enables the use of an object-oriented configuration methodology by the process engineer, which is intuitive and easy to use. Configuration time can be reduced by means of bulk engineering functions such as the import/export of process objects and creation of templates using standard Microsoft Office tools such as Excel.

This capability enables greater engineering efficiency, helps to avoid configuration errors, and, therefore, enhances productivity in engineering.

Automatic change management and copying of process areas

With PCS 7, complex process areas, units, or entire production lines can be copied/cloned quickly with minimal retesting required. For instance, a configured and tested unit can be copied in its entirety, including all its CFCs, SFCs, pictures, scripts, and archive parameters. This function allows a fully configured and tested unit, consisting of CFCs, SFCs, pictures, faceplates, scripts, archive parameters etc., to be copied or modified by automatically updating the links and connections between each element. It offers enormous cost-saving potential for engineering and for validation, especially for plants with repetitive structures.

Reduced turnaround times

Modifications can now be propagated throughout the system faster and while online with PCS 7. Adding a new process object, such as a PID control loop, takes only a few minutes. This minimizes waiting time and helps lower the costs of commissioning and operation. *

Automatic optimization of control strategy

The system automatically optimizes the execution (scan) sequence of the CFCs, SFCs and FBs contained in the user program. This saves time, improves the effectiveness of the control strategy, and significantly improves resource availability in the controller.

Version cross checker supports plant validation

The version cross checker tool guides the user to quickly distinguish and comprehend the differences between two versions of a project.

start | Reduced Engineering ... | 4:21 PM

D-3

= EXHIBIT E =

PID Control - Practical Process Control Training - Microsoft Internet Explorer provided by Comcast

http://www.bin95.com/PID_Process_Control_Saint-Louis.htm

Saint Louis - Phase II Process Control Training

Practical Process Control Training - PID Control

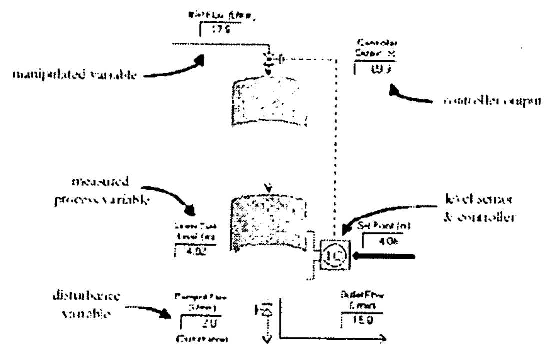
Industrial training services offered by BIN in conjunction with Control Station
Saint Louis, MO. Schedule: Mar. 28-29, 2007
Early bird price $950 (register now)
Or see our Atlanta, GA PID Course! All training courses offered as On-Site Training too

Day 1 Training ...

Lecture
- Fundamentals of Process Dynamics and Control
- Proportional Control
- Integral Action and PI Control
- A Formal Approach to Controller Design Using Design Tools
- Controller Performance Criteria
- Derivative Mode and PID Control
- PID Control with Derivative Filter

Demonstration
- Modeling Process Dynamics Using Control Station
- Implementation of P-Only Controllers
- Adaptive PI Control of Nonlinear Processes
- PID Control of Tank Level
- PID with Filter Control of Heat Exchanger Temperature

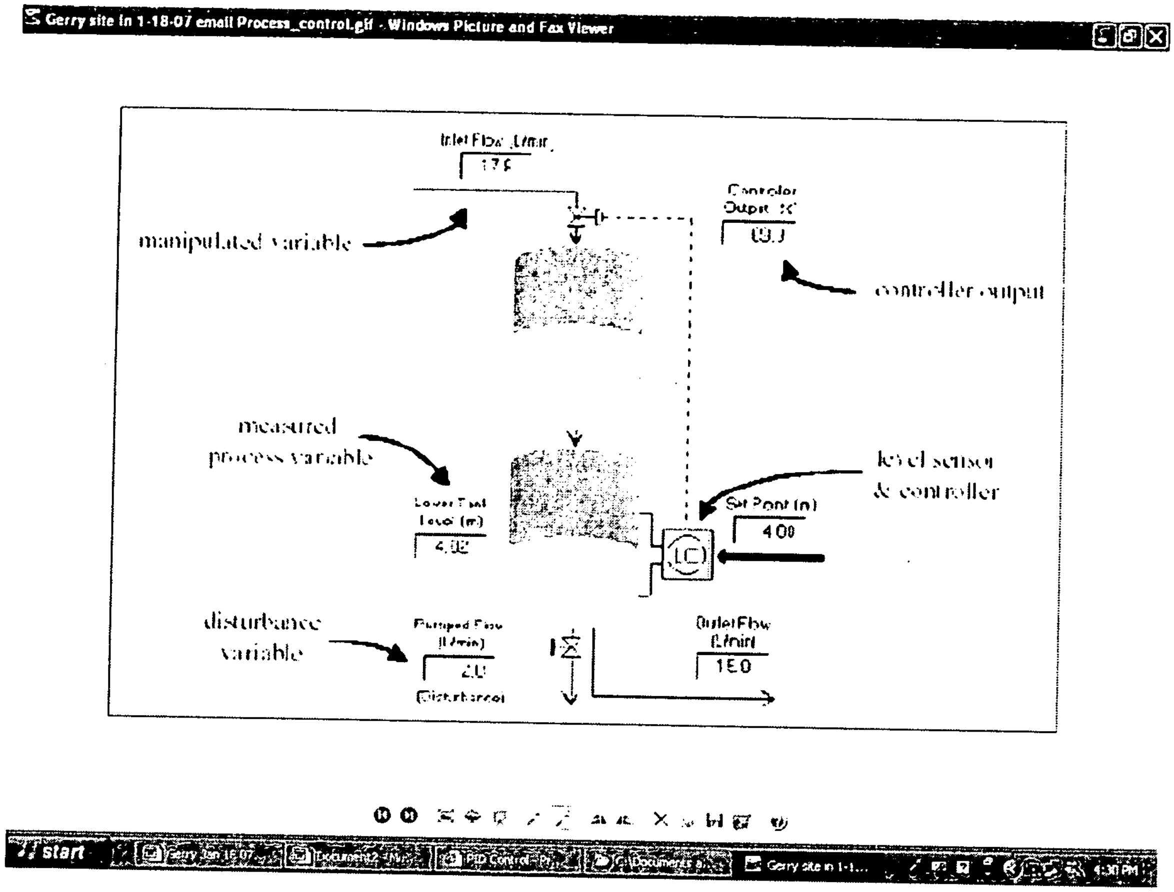
Workshop
- Exploring Dynamics of the Gravity Drained Tanks
- P-Only Control of Tank Level



The **Practical Process Control Training course** (more than just a PID tutorial) will give you a firm foundation in **Process Control and PID control tuning**.

start PID Control - Practica... 4:25 PM

E-1



Gerry site in 1-18-07 email Process_control.gif - Windows Picture and Fax Viewer
manipulated variable
controller output
measured process variable
level sensor & controller
Set Point (m)
4.00
disturbance variable
start

E-2

= EXHIBIT F =

Control.com. nerds in control - Microsoft Internet Explorer provided by Comcast

File Edit View Favorites Tools Help

http://www1.control.com/PLCArchive

control.com®

nerds in control

Web enabled control. One vendor.

Guest

Log in

Join

The Puffin Projects

Search

Links

Topics

About Us

Poetry

Home

The PLCArchive

Make your PLC programming examples part of this community resource. We accept PDF files; contact the maintainer if you need conversion help.

Name	Last Modified	Size
Blink-LED.pdf	Thu May 1 10:29:01 2003	9.55 KB
Controller.pdf	Thu May 1 10:28:37 2003	21.3 KB
Example for using FB41 PID-Control for the SIMATIC S7 PLC's S7-300/400. Copy the data out of the PDF-file directly to a new file in the "Source"-container of STEP7, compile it and your controller is ready to work.	PID Control FB41 STEP7	

)*

)*

start Control.com, nerds in... 4:35 PM

F-1

```
DATA_BLOCK DB 41
TITLE =
AUTHOR : SIMATIC
FAMILY : ICONT
VERSION : 0.0

FB 41
BEGIN
   COM_RST := FALSE;
   MAN_ON := TRUE;
   PVPER_ON := FALSE;
   P_SEL := TRUE;
   I_SEL := TRUE;
   INT_HOLD := FALSE;
   I_ITL_ON := FALSE;
   D_SEL := FALSE;
   CYCLE := T#1S;
   SP_INT := 0.000000e+000;
   PV_IN := 0.000000e+000;
   PV_PER := W#16#0;
   MAN := 0.000000e+000;
   GAIN := 2.000000e+000;
   TI := T#20S;
   TD := T#10S;
   TM_LAG := T#2S;
   DEADB_W := 0.000000e+000;
   LMN_HLM := 1.000000e+002;
   LMN_LLM := 0.000000e+000;
   PV_FAC := 1.000000e+000;
   PV_OFF := 0.000000e+000;
   LMN_FAC := 1.000000e+000;
   LMN_OFF := 0.000000e+000;
   I_ITLVAL := 0.000000e+000;
   DISV := 0.000000e+000;
   LMN := 0.000000e+000;
   LMN_PER := W#16#0;
   QLMN_HLM := FALSE;
   QLMN_LLM := FALSE;
   LMN_P := 0.000000e+000;
   LMN_I := 0.000000e+000;
   LMN_D := 0.000000e+000;
   PV := 0.000000e+000;
   ER := 0.000000e+000;
   sInvAlt := 0.000000e+000;
   sIanteilAlt := 0.000000e+000;
   sRestInt := 0.000000e+000;
   sRestDif := 0.000000e+000;
   sRueck := 0.000000e+000;
   sLmn := 0.000000e+000;
   sbArwHLmOn := FALSE;
   sbArwLLmOn := FALSE;
   sbILimOn := TRUE;
END_DATA_BLOCK


ORGANIZATION_BLOCK OB 35
TITLE = "Cyclic Interrupt"
VERSION : 0.1

VAR_TEMP
  OB35_EV_CLASS : BYTE ;          //Bits 0-3 = 1 (Coming event), Bits 4-7 = 1
(Event class 1)
  OB35_STRT_INF : BYTE ;          //16#36 (OB 35 has started)
  OB35_PRIORITY : BYTE ;          //Priority of OB Execution
  OB35_OB_NUMBR : BYTE ;          //35 (Organization block 35, OB35)
  OB35_RESERVED_1 : BYTE ;        //Reserved for system
  OB35_RESERVED_2 : BYTE ;        //Reserved for system
```

```
   OB35_PHASE_OFFSET : WORD ;    //Phase offset (msec)
   OB35_RESERVED_3 : INT ;       //Reserved for system
   OB35_EXC_FREQ : INT ; //Frequency of execution (msec)
   OB35_DATE_TIME : DATE_AND_TIME ;      //Date and time OB35 started
END_VAR
BEGIN
NETWORK
TITLE =Wenn switching from Auto->Hand then takeover output on Handvalue
//Flankdetection on Automatik to Manual
      A     M     10.0; // If hand is choossen then switch the actual output on
the hand value
      FP    M     11.0; //  one time !!, so a positive flank
      =     M     11.1;
      JCN   cont; // When no flank then to the controller

      L     MD    48; // Output
      T     MD    20; // --> Manual value
NETWORK
TITLE =Controller
//Not all the parameters are necessary on the FB, here this is done to test it
//easy in a VAT-table.
//
//The values from the Analog Input and the Analog Output are here 27648
bitvalues
//are 100%. When that is not wanted, the parameters "PV_FAC" for the Process
//Value and "LMN_FAC" for the Output must changed.
//With the parameters "PV_OFF" and "LMN_OFF" the underlimit has to be changed
//wenn that is not zero.
//
cont: CALL FB    41 , DB    41 (
           COM_RST                    := FALSE,// Totall Reset
           MAN_ON                     := M     10.0,// For switching to handvalue
           PVPER_ON                   := TRUE,// Read in ProcesValue over Analog
Input on "PV_VER"
           P_SEL                      := TRUE,// P-Action ON
           I_SEL                      := M     10.1,// I-Action ON
           INT_HOLD                   := M     10.2,// The output of the
integrator can be "frozen" by setting the input "integral action hold."
           I_ITL_ON                   := FALSE,// INITIALIZATION OF THE INTEGRAL
ACTION
           D_SEL                      := M     10.3,// DERIVATIVE ACTION ON
           CYCLE                      := T#1S,// The time between the block calls
must be constant. The "sampling time" input specifies the time between block
calls.
           SP_INT                     := MD    12,// Setpoint
           PV_IN                      := MD    16,// Is switched OFF with PVPER_ON
= TRUE
           PV_PER                     := PIW  512,// Value from Analog Input (MW4)
           MAN                        := MD    20,// The "manual value" input is
used to set a manual value using the operator interface functions.
           GAIN                       := MD    24,
           TI                         := MD    28,
           TD                         := MD    32,
           DEADB_W                    := MD    36,// A dead band is applied to the
error. The "dead band width" input determines the size of the dead band.
           LMN_HLM                    := MD    40,// The manipulated value is
always limited by an upper and lower limit. The "manipulated value high limit"
input specifies the upper limit.
           LMN_LLM                    := MD    44,// The manipulated value is
always limited by an upper and lower limit. The "manipulated value low limit"
input specifies the lower limit.
           PV_FAC                     := 1.000000e+000,// = Standard, scaling
Proces Value between 0...100% = 0...27648
           PV_OFF                     := 0.000000e+000,// no Offset
           LMN_FAC                    := 1.000000e+000,// = Standard, scaling
Output Value between 0...100% = 0...27648
           LMN_OFF                    := 0.000000e+000,// no Offset
           LMN                        := MD    48,// The effective manipulated
```

```
value is output in floating point format at the "manipulated value" output.
            LMN_PER                        := PQW  512,// (MW6) The manipulated value
in the I/O format is connected to the controller at the "manipulated value
peripheral" output.
            QLMN_HLM                       := M      10.4,
            QLMN_LLM                       := M      10.5,
            LMN_P                          := MD     52,
            LMN_I                          := MD     56,
            LMN_D                          := MD     60,
            PV                             := MD     64,
            ER                             := MD     68);
      BE    ;
END_ORGANIZATION_BLOCK

ORGANIZATION_BLOCK OB 100
TITLE = "Complete Restart"
VERSION : 0.1


VAR_TEMP
  OB100_EV_CLASS : BYTE ;        //16#13, Event class 1, Entering event state,
Event logged in diagnostic buffer
  OB100_STRTUP : BYTE ; //16#81/82/83/84 Method of startup
  OB100_PRIORITY : BYTE ;        //Priority of OB Execution
  OB100_OB_NUMBR : BYTE ;        //100 (Organization block 100, OB100)
  OB100_RESERVED_1 : BYTE ;      //Reserved for system
  OB100_RESERVED_2 : BYTE ;      //Reserved for system
  OB100_STOP : WORD ;    //Event that caused CPU to stop (16#4xxx)
  OB100_STRT_INFO : DWORD ;      //Information on how system started
  OB100_DATE_TIME : DATE_AND_TIME ;      //Date and time OB100 started
END_VAR
BEGIN
NETWORK
TITLE =Place first values to the controller

      OPN   DB     41;
      SET   ;
      =     M      10.0;
      =     M      10.1;

      L     1.000000e+001;
      T     MD     12;
      L     2.000000e+000;
      T     MD     24;
      L     T#20S;
      T     MD     28;
      L     T#10S;
      T     MD     32;
      L     1.000000e+002;
      T     MD     40;
      BE    ;
END_ORGANIZATION_BLOCK
```

= EXHIBIT G =

Single-loop controllers and PLC - Microsoft Internet Explorer provided by Comcast

File Edit View Favorites Tools Help

https://www.control.com/thread/1026228393

control.com

nerds in control

Welcome to Control.com, the global online community of automation professionals.

from the Automation List department...

Single-loop controllers and PLC

Posted by Anonymous on 11 November, 2006 - 2:20 am

Which one is better to design a process plant which has several PID controlling loops

1) Connect all loops to the plant PLC, do some PID programming, and let the PLC control the loops directly

2) Utilize single-loop controllers (with PID capability) for every single loop, then integrate them to the plant PLC using communication scheme such as MODBUS, etc.

Reply to this post...

Posted by Marcos Thoni Bergamo on 14 November, 2006 - 10:58 pm

Mr. Anonymous,

Some positive points for the PLC solution:
You may have a faster PID update cicle (compared to low-cost single loops). This may be important to flow and other fast loops
You may have a better "non-continuous" conditions management, including some logic to treat special batch conditions

Some positive points for the single loops.
Reliability: a failure in one controller will not stop all the system
. Maintenance, every controller have one input, one output, display, keyboard, and you don't need special cable or software to program/debug it. An instrumentist can fix a problem at saturday 3AM

start | Sorry Jan 18 07 email... | C:\Documents and Se... | Single-loop controller ... | 4:40 PM

G-1

from the Automation List department...

Single-loop controllers and PLC

Posted by Anonymous on 11 November, 2006 - 2:20 am

Which one is better to design a process plant which has several PID controlling loops:

1) Connect all loops to the plant PLC, do some PID programming, and let the PLC control the loops directly.

2) Utilise single-loop controllers (with PID capability) for every single loop, then integrate them to the plant PLC using communication scheme such as MODBUS, etc.

Posted by Marcos Thoni Bergamo on 14 November, 2006 - 10:58 pm

Mr. Anonymous,

Some positive points for the PLC solution:
. You may have a faster PID update cicle (compared to low-cost single loops). This may be important to flow and other fast loops.
. You may have a better "non-continuous" conditions management, including some logic to treat special batch conditions.

Some positive points for the single loops:
. Reliability: a failure in one controller will not stop all the system
. Maintenance: every controller have one input, one output, display, keyboard, and you don't need special cable or software to program/debug it. An instrumentist can fix a problem at saturday 3AM.

Regards

Marcos T. B.

Posted by Anonymous on 6 December, 2006 - 9:20 pm

I have replaced single loop controlers with PLC's for years. I need to switch between different controlers during the process.I also pass values and control bits between loops. The input and output configurations vary but I maintain a structured data block for each controler. This makes HMI screens and historical base management easy for multiple loops. With newer PLC's this is also possible with multi-instance processing.In many cases the HMI database can be populated straight from the PLC. I am now starting a project with S-7 and IFIX and am searching for peoples experience with Siemens PID controlers. In the past I have used other brands or written my own.

Posted by Steve Hartigan on 8 December, 2006 - 7:52 pm

Siemens purchased Moore Automation over five years ago. They still market their 353 Controllers which I found to be the most robust and advanced devices available. That said, they are a bit pricier than some others on the market. At an LNG plant that I previously worked at, they were the sole control platform on several heat exhangers and boiler controls.

I was recently told that Emerson (Fisher-Rosemount) is no longer making their model. I have heard good things about the Foxboro products, but have never seen them in action. I hope that this helps.

Posted by Dennis Patterson on 8 December, 2006 - 9:42 pm

I would suggest using loops in the PLC, it is more cost effective, easy to do and scalable. In the Boral cement plant i work at, i add and remove loops all day. If the process engineer wants to cascade a temperature and flow loop its simple to perform in the PLC without tedious wiring and saves days of work at a time. We have hundreds of loops running in the PLC to SCADA, using S7 and TI with CITECT. If you want an example just e-mail me and ill send you a S7 example.

automation-integrator@iinet.net.au

From Control Engineering magazine...

6-2

Related articles from Control Engineering magazine

- Doing flexible manufacturing right with S88 design
- Product lifecycle: Siemens moves to acquire UGS Corp.
- Distribution: Microchip Technology development software available via download
- Bus & Board 2007: New name, new products introduced
- Bus & Board 2007: Speakers address VMEbus trends, needs, new technologies
- Products: Controller trouble-shooting software on site or long distance
- Products: Mass flowmeter/controller gives fast response in gas applications
- Products: New alarm notification system upgrades features
- News: Use of collaborative production management for process manufacturing growing quickly
- Opinion: Advanced control schemes aren't just for big guys

6-3

= EXHIBIT H =

SOUTH COAST SYSTEMS, INC.

PLC Programming

South Coast Systems has 25 years experience in PLC code development specializing in Allen-Bradley, GE, Modicon, and TI/Siemens PLC systems. We can assist you in the development of your process description, piping & instrumentation diagrams - PID's (for complete CAD design services go to Engineering Services), loop drawings, and flow charts.

With the standards SCS has developed over the years for motor, valve, and PID control, our objective is to provide code that your technicians will be able to understand enabling them to troubleshoot problems on their own while minimizing process downtime.

*

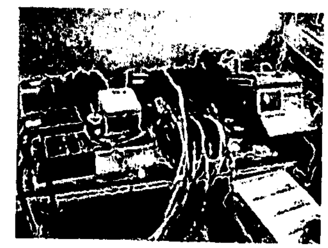
If you have any questions or a problem that we may be of assistance please send an email: info@southcoastsystems.com or call 714 834-0232.

~~#~~ H-1

= EXHIBIT I =

Programmable logic controller

From Wikipedia, the free encyclopedia


PLC & input/output arrangements

A **Programmable Logic Controller**, **PLC**, or **Programmable Controller** is an electronic device used for automation of industrial processes, such as control of machinery on factory assembly lines. Unlike general-purpose computers, the PLC is designed for extended temperature ranges, dirty or dusty conditions, immunity to electrical noise, and resistance to vibration and impact. Programs to control machine operation are typically stored in battery-backed or read-only memory. A PLC is an example of a real time system since output results must be produced in response to input conditions within a bounded time, otherwise miscontrol will result.

Contents

7-1

Features

The main difference from other computers are the special input/output arrangements. These connect the PLC to sensors

and actuators. PLCs read limit switches, analog process variables (such as temperature and pressure), and the positions of complex positioning systems. Some even use machine vision. On the actuator side, PLCs operate electric motors, pneumatic or hydraulic cylinders or diaphragms, magnetic relays or solenoids, or analog outputs. The input/output arrangements may be built into a simple PLC, or the PLC may have external I/O modules attached to a computer network that plugs into the PLC.

PLCs were invented as replacements for automated systems that would use hundreds or thousands of relays, cam timers, and drum sequencers. Often, a single PLC can be programmed to replace thousands of relays. Programmable controllers were initially adopted by the automotive manufacturing industry, where software revision replaced the re-wiring of hard-wired control panels when production models changed.

Many of the earliest PLCs expressed all decision making logic in simple ladder logic which appeared similar to electrical schematic diagrams. The electricians were quite able to trace out circuit problems with schematic diagrams using ladder logic. This program notation was chosen to reduce training demands for the existing technicians. Other early PLCs used a form of instruction list programming, based on a stack-based logic solver.

The functionality of the PLC has evolved over the years to include sequential relay control, motion control, process control, distributed control systems and networking. The data handling, storage, processing power and communication capabilities of some modern PLCs are approximately equivalent to desktop computers. PLC-like programming combined with remote I/O hardware, allow a general-purpose desktop computer to overlap some PLCs in certain applications. 7-2

Under the IEC 61131-3 standard, PLCs can be programmed using standards-based programming languages. A graphical programming notation called Sequential Function Charts is available on certain programmable controllers.

PLC compared with other control systems

PLCs are well-adapted to a certain range of automation tasks. These are typically industrial processes in manufacturing where the cost of developing and maintaining the automation system is high relative to the total cost of the automation, and where changes to the system would be expected during its operational life. PLCs contain input and output devices compatible with industrial pilot devices and controls; little electrical design is required, and the design problem centers on expressing the desired sequence of operations in ladder logic (or function chart) notation. PLC applications are typically

highly customized systems so the cost of a packaged PLC is low compared to the cost of a specific custom-built controller design. On the other hand, in the case of mass-produced goods, customized control systems are economic due to the lower cost of the components, which can be optimally chosen instead of a "generic" solution, and where the non-recurring engineering charges are spread over thousands of sales.

Some modern PLCs with full capabilities are available for a few hundred USD. This allows them to be economically applied on very small control problems.

For high volume or very simple fixed automation tasks, different techniques are used. For example, a consumer dishwasher would be controlled by an electromechanical cam timer costing only a few dollars in production quantities.

A microcontroller-based design would be appropriate where hundreds or thousands of units will be produced and so the development cost (design of power supplies and input/output hardware) can be spread over many sales, and where the end-user would not need to alter the control. Automotive applications are an example; millions of units are built each year, and very few end-users alter the programming of these controllers. (However, some specialty vehicles such as transit busses economically use PLCs instead of custom-designed controls, because the volumes are low and the development cost would be uneconomic.)

I-3

Very complex process control, such as used in the chemical industry, may require algorithms and performance beyond the capability of even high-performance PLCs. Very high speed controls may also require customised solutions; for example, aircraft flight controls.

PLCs may include logic for single-variable feedback analog control loop, a "proportional, integral, derivative" or "PID controller." A PID loop could be used to control the temperature of a manufacturing process, for example. Historically PLCs were usually configured with only a few analog control loops; where processes required hundreds or thousands of loops, a distributed control system (DCS) would instead be used. However, as PLCs have become more powerful, the boundary between DCS and PLC applications has become less clear-cut.

Digital and analog signals

Digital or discrete signals behave as switches, yielding simply an On or Off signal (1 or 0, True or False, respectively).

Pushbuttons, limit switches, and photo-eyes are examples of devices providing a discrete signal. Discrete signals are Sent using either voltage or current, where a specific range is designated as *On* and another as *Off*. A PLC might use 24 V DC I/O, with values above 22 V DC representing *On* and values below 2VDC representing *Off*. Initially, PLCs had only discrete I/O.

Analog signals are like volume controls, with a range of values between zero and full-scale. These are typically interpreted as integer values (counts) by the PLC, with various ranges of accuracy depending on the device and the number of bits available to store the data. Pressure, temperature, flow, and weight are often represented by analog signals. Analog signals can use voltage or current with a magnitude proportional to the value of the process signal. For example, an analog 4-20 mA or 0 - 10 V input would be converted into an integer value of 0 - 32767.

Current inputs are less sensitive to electrical noise (i.e. from welders or electric motor starts) than voltage inputs.

Example

As an example, say the facility needs to store water in a tank. The water is drawn from the tank by another system, as needed and our example system must manage the water level in the tank.

I-4

Using only digital signals, the PLC has two digital inputs from float switches (tank empty and tank full). The PLC uses a digital output to open and close the inlet valve into the tank.

If both float switches are off (down) or only the 'tank empty' switch is on, the PLC will open the valve to let more water in. If only the 'tank full' switch is on, the valve turns off. Both switches being on would signal that something is wrong with one of the switches, as the tank cannot be both full and empty at the same time. Two float switches are used to prevent a 'flutter' condition where any water usage activates the pump for a very short time causing the system to wear out faster.

An analog system might use a load cell (scale) that weighs the tank, and an adjustable (throttling) valve. The PLC could use a PID feedback loop to control the valve opening. The load cell is connected to an analog input and the valve is connected to an analog output. This system fills the tank faster when there is less water in the tank. If the water level drops rapidly, the valve can be opened wide. If water is only dripping out of the tank, the valve adjusts to slowly drip

water back into the tank.

In this system, to avoid 'flutter' adjustments that can wear out the valve, many PLCs have a "deadband". A technician adjusts this deadband so the valve moves only for a significant change in rate. This will in turn minimize the motion of the valve, and reduce its wear.

A real system might combine both approaches, using float switches and simple valves to prevent spills, and a rate sensor and rate valve to optimize refill rates. Backup and maintenance methods can make a real system very complicated.

PLCs package will work on I/O capabilities: Modular, Rack, P2P

Modular PLCs have a limited number of connections built in for inputs and outputs. Typically, expansions are available if the base model does not have enough I/O.

Rack-style PLCs have processor modules with separate [optional] I/O modules, which may occupy many racks. These often have thousands of discrete and analog inputs and outputs. Often a special high speed serial I/O link is used so that racks can be remotely mounted from the processor, reducing the wiring costs for large plants. Some of today's PLCs can communicate over a wide range of media including RS-485, Coaxial, and even Ethernet for I/O control at network speeds up to 100Mbps. I-5

PLCs used in larger I/O systems may have peer-to-peer (P2P) communication between processors. This allows separate parts of a complex process to have individual control while allowing the subsystems to co-ordinate over the communication link. These communication links are also often used for HMI devices such as keypads or PC-type workstations.

A rule-of thumb is that the average number of inputs installed is three times that of outputs for both analog and digital. The 'extra' inputs arise from the need to have redundant methods to monitor an instrument to appropriately control another, and from the need to use both manual command inputs to the system and feedback from the controlled system itself.

Programming

Early PLCs, up to the mid-1980s, were programmed using proprietary programming panels or special-purpose programming terminals, which often had dedicated function keys representing the various logical elements of PLC programs. Programs were stored on cassette tape cartridges. Facilities for printing and documentation were very minimal due to lack of memory capacity. More recently, PLC programs are typically written in a special application on a personal computer, then downloaded by a direct-connection cable or over a network to the PLC. The very oldest PLCs used non-volatile magnetic core memory but now the program is stored in the PLC either in battery-backed-up RAM or some other non-volatile flash memory.

Early PLCs were designed to be used by electricians who would learn PLC programming on the job. These PLCs were programmed in "ladder logic", which strongly resembles a schematic diagram of relay logic. Modern PLCs can be programmed in a variety of ways, from ladder logic to more traditional programming languages such as BASIC and C. Another method is State Logic, a Very High Level Programming Language designed to program PLCs based on State Transition Diagrams.

I-6

Recently, the International standard IEC 61131-3 has become popular. IEC 61131-3 currently defines five programming languages for programmable control systems: FBD (Function block diagram), LD (Ladder diagram), ST (Structured text, similar to the Pascal programming language), IL (Instruction list, similar to assembly language) and SFC (Sequential function chart). These techniques emphasize logical organization of operations.

While the fundamental concepts of PLC programming are common to all manufacturers, differences in I/O addressing, memory organization and instruction set mean that PLC programs are never perfectly interchangeable between different makers. Even within the same product line of a single manufacturer, different models may not be directly compatible.n

User interface

PLCs may need to interact with people for the purpose of configuration, alarm reporting or everyday control. A Human-Machine Interface (HMI) is employed for this purpose.

A simple system may use buttons and lights to interact with the user. Text displays are available as well as graphical touch screens. Most modern PLCs can communicate over a network to some other system, such as a computer running a SCADA (Supervisory Control And Data Acquisition) system or web browser.

Communications

PLCs usually have built in communications ports for at least RS232, and optionally for RS485 and ethernet. Modbus is the lowest common denominator communications protocol. Others are various fieldbuses such as Profibus. Other communications protocols that may be used are listed in the List of automation protocols.

History

I-7

The PLC was invented in response to the needs of the American automotive industry. Before the PLC, control, sequencing, and safety interlock logic for manufacturing automobiles was accomplished using relays, timers and dedicated closed-loop controllers. The process for updating such facilities for the yearly model change-over was very time consuming and expensive, as the relay systems needed to be rewired by skilled electricians. In 1968 GM Hydramatic (the automatic transmission division of General Motors) issued a request for proposal for an electronic replacement for hard-wired relay systems.

The winning proposal came from Bedford Associates of Bedford, Massachusetts. The first PLC, designated the 084 because it was Bedford Associates eighty-fourth project, was the result. Bedford Associates started a new company dedicated to developing, manufacturing, selling, and servicing this new product: Modicon, which stood for MOdular DIgital CONtroller. One of the people who worked on that project was Dick Morley, who is considered to be the "father" of the PLC. The Modicon brand was sold in 1977 to Gould Electronics, and later acquired by German Company AEG and then by Schneider Electric, the current owner.

One of the very first 084 models built is now on display at Modicon's headquarters in North Andover, Massachusetts. It was presented to Modicon by GM, when the unit was retired after nearly twenty years of uninterrupted service.

The automotive industry is still one of the largest users of PLCs, and Modicon still numbers some of its controller models such that they end with eighty-four. PLCs are used in many different industries and machines such as packaging and semiconductor machines. Well known PLC brands are ABB Ltd., Koyo, Honeywell, Siemens, Modicon, Omron, Allen-Bradley, General Electric, Tesco Controls, Panasonic (Matsushita), and Mitsubishi.

See also

- IEC-61131-3
- Fieldbus
- Simatic S5 PLC

External links

- PLC tutorial (http://www.plcs.net/)
- Interview with Dick Morley (pdf) (http://www.ccontrols.com/pdf/Extv4n2.pdf)
- Timeline of PLC History (http://www.plcdev.com/plc_timeline)

I-8

Retrieved from "http://en.wikipedia.org/wiki/Programmable_logic_controller"

Categories: Digital electronics | Embedded systems | Production and manufacturing | Automation | Control Engineering

---

Distributed control system

= EXHIBIT J =

From Wikipedia, the free encyclopedia
(Redirected from Distributed Control System)

A **distributed control system** (DCS) refers to a control system usually of a manufacturing system or process or any kind of dynamic system, in which the controller elements are not central in location (like the brain) but are distributed throughout the system with each component sub-system under the control of one or more controllers. The entire system may be networked for communication and monitoring.

Distributed control systems (DCSs) are used in industrial, electrical, computer and civil engineering applications to monitor and control distributed equipment with or without remote human intervention; the nomenclature for the former 'manual control' and the latter 'automated control'.

A DCS typically uses computers (usually custom designed processers) as controllers and use both proprietary interconnections and protocols for communication. Input & output modules form component parts of the DCS. The processor (which is a part of the controller) receives information from input modules and sends information to output modules. The input modules receive information from input instruments in the process (aka field) and output modules transmit to the output instruments in the field. Computer buses or electrical buses connect the processor and modules through multiplexers/demultiplexers. They also connect the distributed controllers with the central controller and finally to the Human-Machine Interface (HMI) or control consoles. See PAS.

DCS is a very broad term that describes solutions across a large variety of industries, including:

- Electrical power grids and electrical generation plants
- Environmental control systems
- Traffic signals
- Water management systems
- Refining and chemical plants
- Pharmaceutical manufacturing
- Sensor Networks

The broad architecture of a solution involves either a direct connection to physical equipment such as switches, pumps and valves or connection via a secondary system such as a SCADA system.

A DCS solution does not require operator intervention for its normal operation, but with the line between SCADA and DCS merging, systems claiming to offer DCS may actually permit operator interaction via a SCADA system.

Distributed Control Systems (DCSs) are dedicated systems used to control manufacturing processes that are continuous or batch-oriented, such as oil refining, petrochemicals, central station power generation, pharmaceuticals, food & beverage manufacturing, cement production, steelmaking, and papermaking. DCSs are connected to sensors and actuators and use setpoint control to control the flow of material through the plant. The most common example is a setpoint control loop consisting of a pressure sensor, controller, and control valve. Pressure or flow measurements are transmitted to the controller, usually through the aid of a signal conditioning Input/Output (I/O) device. When the measured variable reaches a certain point, the controller instructs a valve or actuation device to open or close until the fluidic flow process reaches the desired setpoint. Large oil refineries have many thousands of I/O points and employ very large DCSs. Processes are not limited to fluidic flow through pipes, however, and can also include things like paper machines and their associated variable speed drives and motor control centers, cement kilns, mining operations and ore processing facilities, and many others.

A typical DCS consists of functionally and/or geographically distributed digital controllers capable of executing from 1 to 256 or more regulatory control loops in one control box. The input/output devices (I/O) can be integral with the controller or located remotely via a field network. Today's controllers have extensive computational capabilities and, in addition to proportional, integral, and derivative (PID) control, can generally perform logic and sequential control.

DCSs may employ one or several workstations and can be configured at the workstation or by an off-line personal computer. Local communication is handled by a control network with transmission over twisted pair, coaxial, or fiber optic cable. A server and/or applications processor may be included in the system for extra computational, data collection, and reporting capability.

History

The DCS was introduced in 1975. Both Honeywell and Japanese electrical engineering firm Yokogawa introduced their own independently produced DCSs at roughly the same time, with the TDC 2000 and CENTUM systems, respectively. US-based Bristol also introduced their UCS 3000 universal controller in 1975. In 1980, Bailey (now part of ABB) introduced the NETWORK 90 system.

The DCS largely came about due to the increased availability of microcomputers and the proliferation of microprocessors in the world of process control. Computers had already been applied to process automation for some time in the form of Set Point Control, where process computers supervised clusters of analog controllers. The proliferation of microprocessors allowed suppliers to take this mode to the next step by deploying minicomputers in a supervisory role, controlling several digital loop controllers. A CRT-based workstation provided visibility into the process using text and crude character graphics. Availability of a fully functional graphical user interface was a long way away.

Central to the DCS model was the inclusion of control function blocks, which were introduced by the Foxboro company. One of the first embodiments of object-oriented software, function blocks were self contained "blocks" of code that emulated analog hardware control components and performed tasks that were essential to process control, such as execution of PID algorithms. Function blocks continue to endure as the predominant method of control for DCS suppliers, and are supported by key technologies such as **Foundation Fieldbus** [1] (http://www.fieldbus.org/) today.

Digital communication between controllers and supervisory computers was one of the primary advantages of the DCS, and attention was duly focused on the networks, which provided the all-important lines of communication that, for process applications, had to incorporate specific functions such as determinism and redundancy. As a result, many suppliers embraced the IEEE 802.4 networking standard. This decision set the stage for the wave of migrations necessary when information technology moved into process automation and IEEE 802.3 rather than IEEE 802.4 prevailed as the control LAN.

The Network Centric Era of the 1980s

The DCS brought distributed intelligence to the plant and established the presence of computers and microprocessors in process control, but it still did not provide the reach and openness necessary to unify plant resource requirements. In many cases, the DCS was merely a digital replacement of the same functionality provided by analog controllers and a

panelboard display. This was embodied in The Perdue Reference Model (PRM) that was developed to define Manufacturing Operations Management relationships. PRM later formed the basis for ISA95 standards activities today.

In the 1980s, users began to look at DCSs as more than just basic process control. It was believed that if openness could be achieved and greater amounts of data could be shared throughout the enterprise that good things could be achieved, although few were sure what these benefits would be. The first attempts to increase the openness of DCSs resulted in the adoption of the predominant operating system of the day -- UNIX. UNIX and its companion networking technology TCP-IP were developed by the Department of Defense for openness, which was precisely the issue the process industries were looking to resolve.

As a result suppliers also began to adopt Ethernet-based networks with their own proprietary protocol layers. The full TCP/IP standard was not implemented, but the use of Ethernet made it possible to implement the first instances of object management and global data access technology. The 1980s also witnessed the first PLCs integrated into the DCS infrastructure. Plant-wide historians also emerged to capitalize on the extended reach of automation systems. The first DCS supplier to adopt UNIX and Ethernet networking technologies was Foxboro, who introduced the I/A Series system in 1987.

The Application Centric Era of the 1990s

The drive toward openness in the 1980s gained momentum through the 1990s with the increased adoption of Commercial-Off-The-Shelf (COTS) components and IT standards. Probably the biggest transition undertaken during this time was the move from the UNIX operating system to the Windows environment. While the realm of the real time operating system (RTOS) for control applications remains dominated by real time commercial variants of UNIX or proprietary operating systems, everything above real-time control has made the transition to Windows.

The invasion of Microsoft at the desktop and server layers resulted in the development of technologies such as OLE for Process Control (OPC), which is now a de facto industry connectivity standard. Internet technology also began to make its mark in automation and the DCS world, with most DCS HMI supporting Internet connectivity. The '90s were also known for the "Fieldbus Wars", where rival organizations competed to define what would become the IEC fieldbus standard for digital communication with field instrumentation instead of 4-20 milliamp analog communications. The first fieldbus installations occurred in the 1990s. Towards the end of the decade, the technology began to develop significant

momentum, with the market consolidated around Foundation Fieldbus and Profibus PA for process automation applications. Some suppliers built new systems from the ground up to maximize functionality with fieldbus, such as Emerson with the DeltaV control system.

The impact of COTS, however, was most pronounced at the hardware layer. For years, the primary business of DCS suppliers had been the supply of large amounts of hardware, particularly I/O and controllers. The initial proliferation of DCSs required the installation of prodigious amounts of this hardware, most of it manufactured from the bottom up by DCS suppliers. Standard computer components from manufacturers such as Intel and Motorola, however, made it cost prohibitive for DCS suppliers to continue making their own components, workstations, and networking hardware.

As the suppliers made the transition to COTS components, they also discovered that the hardware market was shrinking fast. COTS not only resulted in lower manufacturing costs for the supplier, but also steadily decreasing prices for the end users, who were also becoming increasingly vocal over what they perceived to be unduly high hardware costs. Some suppliers that were previously stronger in the PLC business, such as Rockwell Automation and Siemens, were able to leverage their expertise in manufacturing control hardware to enter the DCS marketplace with cost effective offerings.

54

To compound the issue, suppliers were also realizing that the hardware market was becoming saturated. The lifecycle of hardware components such as I/O and wiring is also typically in the range of 15 to over 20 years, making for a challenging replacement market. Many of the older systems that were installed in the 1970s and 1980s are still in use today, and there is a considerable installed base of systems in the market that are approaching the end of their useful life. Developed industrial economies in North America, Europe, and Japan already had many thousands of DCSs installed, and with few if any new plants being built, the market for new hardware was shifting rapidly to smaller, albeit faster growing regions such as China, Latin America, and Eastern Europe.

Because of the shrinking hardware business, suppliers began to make the challenging transition from a hardware-based business model to one based on software and value-added services. It is a transition that is still being made today. The applications portfolio offered by suppliers expanded considerably in the '90s to include areas such as production management, model-based control, real-time optimization, Plant Asset Management (PAM), Real Time Performance Management (RPM) tools, alarm management, and many others. To obtain the true value from these applications, however, often requires a considerable service content, which the suppliers also provide. DCS supplier services have also expanded in scope to the point where many suppliers can act as Main Automation Contractors (MACs), providing a

single point of responsibility for all automation-related facets of a project.

See also

- SCADA
- PLC

External links

Retrieved from "http://en.wikipedia.org/wiki/Distributed_control_system"

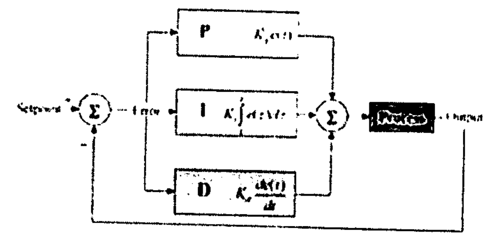
Category: Control Engineering

PID controller

= EXHIBIT K =

From Wikipedia, the free encyclopedia

A **proportional-integral-derivative controller (PID controller)** is a common feedback loop component in industrial control systems (see also control theory).



A traditional PID controller

The controller takes a measured value from a process or other apparatus and compares it with a reference setpoint value. The difference (or "error" signal) is then used to adjust some input to the process in order to bring the process' measured value back to its desired setpoint. Unlike simpler controllers, the PID can adjust process outputs based on the history and rate of change of the error signal, which gives more accurate and stable control. In contrast to more complex algorithms such as optimal control theory, PID controllers can often be adjusted without advanced mathematics. However, pushing robustness and performance to the limits requires a good understanding of the theory and controlled process.

K-1

Contents

Control loop basics

Intuitively, the PID loop tries to automate what an intelligent operator with a gauge and a control knob would do. The operator would read a gauge showing the output measurement of a process, and use the knob to adjust the input of the process (the "action") until the process's output measurement stabilizes at the desired value on the gauge. In older control literature this adjustment process is called a "reset" action. The position of the needle on the gauge is a "measurement", "process value" or "process variable". The desired value on the gauge is called a "setpoint" (also called "set value"). The difference between the gauge's needle and the setpoint is the "error".

A control loop consists of three parts:

1. Measurement by a sensor connected to the process (or the "plant"),
2. Decision in a controller element,
3. Action through an output device ("actuator") such as a control valve.

K-2

As the controller reads a sensor, it subtracts this measurement from the "setpoint" to determine the "error". It then uses the error to calculate a correction to the process's input variable (the "action") so that this correction will remove the error from the process's output measurement.

In a PID loop, correction is calculated from the error in three ways: cancel out the current error directly (Proportional), the amount of time the error has continued uncorrected (Integral), and anticipate the future error from the rate of change of the error over time (Derivative).

For example: suppose a water tank is used to supply water for use in several parts of a plant, and it is necessary to keep the water level constant. A sensor would measure the height of water in the tank, producing the "measurement", and continuously feed this data to the controller. The controller would have a "setpoint" of (for example) 75% full. The controller would have its output (the "action") connected to a proportionally-controlled characterized control valve controlling the make-up water feed. Opening the valve would increase the rate of water entering the tank, closing the

valve would decrease it. The controller would use the measurement of how the level is changing over time to calculate how to manipulate the control valve to maintain a constant level at the "setpoint".

A PID controller can be used to control any measurable variable which can be affected by manipulating some other process variable. For example, it can be used to control temperature, pressure, flow rate, chemical composition, speed, or other variables. Automobile cruise control is an example of a process outside of industry which utilizes crude PID control.

Some control systems arrange PID controllers in cascades or networks. That is, a "master" control produces signals used by "slave" controllers. One common situation is motor controls: one often wants the motor to have a controlled speed, with the "slave" controller (often built into a variable frequency drive) directly managing the speed based on a proportional input. This "slave" input is fed by the "master" controllers' output, which is controlling based upon a related variable.

Coupled and cascaded controls are common in chemical process control, heating, ventilation, and air conditioning systems, and other systems where many parts cooperate.

Theory

K-3

Differing terms are used in the process control industry: The "process variable" is also called the "process's input" or "controller's output." The process's output is also called the "measurement".

The error is found by subtracting the measured quantity from the setpoint.

"PID" is named after its three correcting terms, whose sum constitutes the output of the PID controller.

1. **Proportional** - To handle the immediate error, the error is multiplied by a constant P (for "proportional"). Note that when the error is zero, a proportional controller's output is zero.
2. **Integral** - To learn from the past, the error is integrated and multiplied by a constant I. Without integral term, a PID controller cannot eliminate error if the process requires a non-null input to produce the desired set-point (e.g. heater when controlling a temperature or electrical motor when controlling a speed).

3. **Derivative** - To anticipate the future, the first derivative (the slope of the error) over time is calculated and multiplied by another constant D.

If the PID parameters (the gains of the proportional, integral and derivative terms) are chosen incorrectly, the controlled process input can be unstable, i.e. its output diverges, with or without oscillations, and is limited only by saturations or breakage.

A PID controller is called a PI, PD, or P controller in the absence of respective control actions. It may be noted that EWMA (Exponential Weighted Moving Average) controller is equivalent to PI controller.

Traditionally, the output of the controller (i.e. the input to the process) is given by

$$\mathrm{Output}(t) = P_{\mathrm{contrib}} + I_{\mathrm{contrib}} + D_{\mathrm{contrib}}$$

where P_{contrib}, I_{contrib}, and D_{contrib} are the feedback contributions from the PID controller, defined below:

$$P_{\mathrm{contrib}} = K_p\, e(t)$$
$$I_{\mathrm{contrib}} = K_i \int_0^t e(\tau)\, d\tau$$
$$D_{\mathrm{contrib}} = K_d \frac{de}{dt}$$

K-4

Where $e(t) = \mathit{Setpoint} - \mathit{Measurement}(t)$ is the error signal, and K_p, K_i, K_d are constants that are used to tune the PID control loop:

1. K_p: **Proportional Gain** - Larger K_p typically means faster response since the larger the error, the larger the feedback to compensate.
2. K_i: **Integral Gain** - Larger K_i implies steady state errors are eliminated quicker. The tradeoff is larger overshoot: any negative error integrated during transient response must be integrated away by positive error before we reach steady state.

3. K_d: **Derivative Gain** - Larger K_d decreases overshoot, but slows down transient response.

Normally the controller is implemented with the K_p gain applied to the $I_{contrib}$, and $D_{contrib}$ terms as well in the following form;

$$\mathrm{Output(t)} = K_p\left(e(t) + K_{ip}\int_0^t e(\tau)\,d\tau + K_{dp}\frac{de}{dt}\right)$$

Where $K_{ip} = \frac{K_i}{K_p}$ and $K_{dp} = \frac{K_d}{K_p}$ in relation to the constants defined above.

Most standard tuning methods, such as Ziegler-Nichols and others, are based on this form, as it reduces interaction. In this form, the K_{ip} and K_{dp} gains relate only to dynamics of the process, and the K_p (proportional gain) relates to the gain of the process.

Often, one deals with discrete time intervals instead of the continuity. Thus, the PID controller may also be dealt with recursively:

$$\mathrm{Output}_n = \mathrm{Output}_{n-1} + (K_p + K_i + K_d)\,e_n - (K_p + 2K_d)\,e_{n-1} + K_d\,e_{n-2}$$

K-5

Parameter nomenclature

The parameters of PID control were originally named after the adjustments on mechanical controllers and so were called proportional band, reset and rate. These quantities are based on the interacting algorithm. Modern practice is to refer to gain, integral gain, and derivative gain since this better matches the usage in digital controllers.

There are several different forms of the PID controller. The terms "interacting" and "non-interacting" are used many ways and can lead to confusion.

- The parallel or "non-interacting" form, where the P, I and D parts of the controller are all given the same error input

in parallel and their output is added together. This allows independent adjustment of the proportional, integral and derivative constants.

- The series or "interacting" form, where the output of each part of the controller is used as the input for another part, so that separate P, D and I controllers are connected together in series. This is effectively how older pneumatic and analog electronic controllers worked. It is the more restricted form of the two.

Note that the most common form of PID controller effectively has the P part in series, with its output feeding the I and D parts in parallel. This mixture of the two forms contributes to the confusion surrounding this terminology.

Gain and proportional band are related but inverse quantities. A controller setting of 100% proportional band means that a 100% change of the error signal (setpoint – process variable) will result in 100% change of the output, which is a gain of 1.0. A 20% proportional band indicates that 20% change in error gives a 100% output change, which is a gain of 5.

The reset and rate values are scaled based on the proportional band of the interacting control algorithm. Reset is measured in minutes to correct the output by the proportional band. Rate is measured in proportional band/minute.

Loop tuning

K-6

"Tuning" a control loop is the adjustment of its control parameters (gain/proportional band, integral gain/reset, derivative gain/rate) to the optimum values for the desired control response. The optimum behavior on a process change or setpoint change varies depending on the application. Some processes must not allow an overshoot of the process variable from the setpoint. Other processes must minimize the energy expended in reaching a new setpoint. Generally stability of response is required and the process must not oscillate for any combination of process conditions and setpoints. Tuning of loops is made more complicated by the response time of the process; it may take minutes or hours for a setpoint change to produce a stable effect. Some processes have a degree of non-linearity and so parameters that work well at full-load conditions don't work when the process is starting up from no-load. This section describes some traditional manual methods for loop tuning.

There are several methods for tuning a PID loop. The choice of method will depend largely on whether or not the loop can be taken "offline" for tuning, and the response speed of the system. If the system can be taken offline, the best tuning method often involves subjecting the system to a step change in input, measuring the output as a function of time, and

using this response to determine the control parameters.

If the system must remain online, one tuning method is to first set the *I* and *D* values to zero. Increase the *P* until the output of the loop oscillates, then the *P* should be left set to be approximately half of that value for a "quarter amplitude decay" type response. Then increase *I* until any offset is correct in sufficient time for the process. However too much *I* will cause instability. Finally, increase *D*, if required, until the loop is acceptably quick to reach its reference after a load disturbance. However too much *D* will cause excessive response and overshoot. A fast PID loop tuning usually overshoots slightly to reach the setpoint more quickly; however, some systems cannot accept overshoot, in which case a "critically damped" tune is required, which will require a *P* setting significantly less than half that of the *P* setting causing oscillation.

Effects of *increasing* parameters

Parameter	**Rise Time**	**Overshoot**	**Settling Time**	**S.S. Error**
K_p	Decrease	Increase	Small Change	Decrease
K_i	Decrease	Increase	Increase	Eliminate
K_d	Small Change	Decrease	Decrease	None

K-7

Another tuning method is formally known as the "Ziegler-Nichols method", introduced by John G. Ziegler and Nathaniel B. Nichols. As in the method above, the *I* and *D* gains are first set to zero. The "P" gain is increased until it reaches the "critical gain" K_c at which the output of the loop starts to oscillate. K_c and the oscillation period P_c are used to set the gains as shown:

Ziegler-Nichols method			
Control Type	K_p	K_i	K_d
P		-	-

	$0.5 \cdot K_c$		
PI	$0.45 \cdot K_c$	$1.2K_p / P_c$	-
PID	$0.6 \cdot K_c$	$2K_p / P_c$	$K_p P_c / 8$

Most modern industrial facilities no longer tune loops using the manual calculation methods shown above. Instead, PID tuning and loop optimization software are used to ensure consistent results. These software packages will gather the data, develop process models, and suggest optimal tuning. Some software packages can even develop tuning by gathering data from reference changes.

Mathematical PID loop tuning induces an impulse in the system, and then uses the controlled system's frequency response to design the PID loop values. In loops with response times of several minutes, mathematical loop tuning is recommended, because trial and error can literally take days just to find a stable set of loop values. Optimal values are harder to find. Some digital loop controllers offer a self-tuning feature in which very small setpoint changes are sent to the process, allowing the controller itself to calculate optimal tuning values.

Other formulas are available to tune the loop according to different performance criteria.

K-8

Limitations

The PID controller algorithm itself has some limitations. In practice most problems arise from instrumentation connected to the controller.

One common problem is "integral windup". It might take too long for the output value to ramp up to the necessary value when the loop first starts up. Sometimes this can be fixed with a more aggressive differential term. Sometimes the loop has to be "preloaded" with a starting output. Another option is to disable the integral function until the measured variable has entered the proportional band.

Some PID loops control a valve or similar mechanical device. Wear of the valve or device can be a major maintenance cost. In these cases, the PID loop may have a "deadband" to reduce the frequency of activation of the mechanical device.

This is accomplished by designing the controller to hold its output steady if the change would be small (within the defined deadband range). The calculated output must leave the deadband before the actual output will change. Then, a new deadband will be established around the new output value.

Another problem with the differential term is that small amounts of noise can cause large amounts of change in the output. Sometimes it is helpful to filter the measurements, with a running average, or a low-pass filter. However, low-pass filtering and derivative control cancel each other out, so reducing noise by instrumentation means is a much better choice. Alternatively, the differential band can be turned off in most systems with little loss of control. This is equivalent to using the PID controller as a *PI* controller.

The proportional and differential terms can also produce undesirable results in systems subjected to instantaneous "step" inputs (such as when a computer changes the setpoint). To avoid this, some PID algorithms incorporate various schemes:

1. **derivative of output** Many industrial PID systems actually measure the differential of the output quantity, which is always continuous (i.e., never has a step function), and usually moves in the same direction as the error.
2. **setpoint weighting** Setpoint weighting uses several setpoints. The errors from the two setpoints are combined to reduce upsets. Some schemes slowly reduce the proportion of error from an "old" setpoint, and increase the proportion of error from a "new" setpoint. Other schemes have multiple setpoints controlled by different outside controllers. The error in the integral term must be the true control error to avoid steady-state control errors. These parameters do not affect the response to load disturbances and measurement noise.

K-9

Digital implementations of a PID algorithm may have limitations owing to the sampling rate of the data, and the limits of internal calculation and precision. For example, very old programmable logic controller (PLC) systems may have used only 12 or 16 bits to represent internal variables. Additionally, some software implementations do not correctly handle internal overflow or extreme values, or may arbitrarily limit the values for the adjustable gain parameters.

Another problem faced with PID controllers is that they are linear. Thus performance of PID controllers in non-linear systems (such as HVAC systems) is variable. Often PID controllers are enhanced through methods such as scheduling or fuzzy logic.

Implementation

A PID loop can be implemented with any physical system that can produce ratiometric behavior and integration.

Software PID loops are the most stable, because they do not wear out, and their expense has been decreasing. PID controller functionality is a common feature of PLCs used by many factories.

A PID controller can also be purchased for industrial uses as a panel-mounted controller. These often control only one or two loops and are still used for small stand-alone systems where a PLC or computer control is unnecessary.

PID controllers, when used alone, can give poor performance when the PID loop gains must be reduced so that the control system does not overshoot, oscillate or "hunt" about the control setpoint value. The control system performance can be improve by combining the PID controller functionality with that of a Feed-Forward control output as described in Control Theory. Any information or intelligence derived from the system state can be "fed forward" or combined with the PID output to improve the overall system performance. The Feed-Forward value alone can often provide a major portion of the controller output. The PID controller can then be used to respond to whatever difference or "error" that remains between the controller setpoint and the feedback value. Since the Feed-Forward output is not a function of the process feedback, it can never cause the control system to oscillate, thus improving the system response and stability. For example, in most motion control systems, in order to accelerate a mechanical load under control, more force or torque is required from the prime mover, motor or actuator. If a velocity loop PID controller is being used to control the speed of the load and command the force or torque being applied by the prime mover, then it is beneficial to take the instantaneous acceleration desired for the load, scale that value appropriately and add it to the output of the PID velocity loop controller. This means that whenever the load is being accelerated or decelerated, a proportional amount of force is commanded from the prime mover regardless of the feedback value. The PID loop in this situation uses the feedback information to effect any increase or decrease of the combined output in order to reduce the remaining difference between the process setpoint and the feedback value. Working together, the combined Feed-Forward open loop controller and closed loop PID controller can provide more responsive, stable and intelligent control systems.

K-10

In the early history of automatic process control the PID controller was implemented as a mechanical device, often energized by compressed air. Mechanical systems (once the cheapest) can use a lever, spring and a mass. Pneumatic controllers were once common, but have been largely replaced by digital electronic controllers.

Electronic analog controllers are now very cheap, and can be made from a solid-state or tube amplifier, a capacitor and a resistance. Electronic analog PID control loops were often found within more complex electronic systems, for example, the head positioning of a disk drive, the power conditioning of a power supply, or even the movement-detection circuit of a modern seismometer. Nowadays, they are replaced with digital controllers implemented with microcontrollers or FPGAs.

External links

PID Tutorials

- What is PID? A Tutorial Overview (http://www.expertune.com/r2.asp?f=Wikipedia&l=tutor.html)
- Introduction to Closed-Loop Control (http://www.netrino.com/Publications/Glossary/PID.html)
- PID Tutorial (http://www.engin.umich.edu/group/ctm/PID/PID.html)
- PID Without a PhD (http://www.embedded.com/2000/0010/0010feat3.htm): a beginner's guide to PID loop theory with sample programming code
- Information, including tutorial, on PID control algorithm (http://www.jashaw.com/pid)

Loop Tuning

K-11

- Tuning loops quickly at start-up (http://www.topcontrol.com/pdf/Tuning%20Loops%20Quickly%20at%20Start-Up.pdf)
- Learn how PID and other process tuning works (http://www.bin95.com/PID_Process_Control_Saint-Louis.htm)

References

- Articles, Whitepapers, and tutorials on PID control (http://www.expertune.com/r2.asp?f=Wikipedia&l=articles.html)
- sci.engr.* FAQ on PID controller tuning (http://www.tcnj.edu/~rgraham/PID-tuning.html)

Simulations

- PID controller laboratory, Java applets for PID tuning (http://www.pidlab.com/)
- Good, basic PID simulation in Excel (http://www.htservices.com/Applications/Process/PID2.htm)

Building a PID Controller

- Shows how to build a PID controller with basic electronic components (http://asl.epfl.ch/research/projects/VtolIndoorFlying/rapports/rapportSemStauffer.pdf) go to page 22
- Parts of a Typical Control System (http://www.industrial-electricity.com/open-and-closed-loop-feedback-systems-2-Parts-Typical-Control-System.html)
- Example application: Adding a PID to an espresso machine. (http://www.thedomesticbarista.com/content/view/17/29/)

Special Topics and PID Control Applications

- Azeotropic Distillation Column Control with Fuzzy Logic (http://www.intellopt.com/FLC.htm)
- Proven Methods and Best Practices for PID Control (http://www.controlguru.com/pages/table.html)

Retrieved from "http://en.wikipedia.org/wiki/PID_controller"

K-12

Categories: Pages needing expert attention from Technology experts | Control theory

---

- This page was last modified 04:20, 10 February 2007.

= EXHIBIT L =

Patent Storm

United States Patent 5697436

Proportional with variable bias batch reactor temperature control system

US Patent Issued on December 16, 1997

Inventor(s)

William Harold Johnson
Frederick William Meckley
Breydon Gregory Morton

Assignee

E. I. Du Pont de Nemours and Company

Application

No. 551980 filed on 1995-11-02

Current US Class

165/254 , 236/78D , 700/8

Field of Search

165/254 , 165/292

Examiners

Primary: William E Wayner

Attorney, Agent or Firm

US Patent References

5153807

Description

BACKGROUND OF THE INVENTION

1. Field of the Invention

This invention relates to an improved method of controlling temperature in a jacketed batch reactor. More specifically but not by way of limitation, the invention relates to a novel control strategy and associated control algorithm that employs master (outer) and slave (inner) control loops in a cascaded arrangement with a master controller process variable used as a variable bias in the control algorithm.

2. Description of the Related Art

L-1

The normal engineering practice to deal with a tempered heating/cooling control problem is to install a cascaded control system. In such a situation the slave controller (i.e., the inner control loop) involves the temperature of the jacket heat-exchange fluid and the master controller (i.e., the outer control loop) involves the temperature of the reactor/reaction. Typically the slave controller uses a proportional integral (PI) control algorithm and controls the temperature of the heating/cooling media flowing through the reactor jacket. The master loop uses a proportional integral derivative control algorithm (PID) and controls reactor temperature. Since the controller tuning required to heat/cool as fast as possible is much different from the tuning required to hold the reactor temperature at set point, a sacrifice is normally required which inhibits control system performance. An alternative method uses a nonlinear control algorithm for the master controller and a proportional integral algorithm for the slave loop. While this method is satisfactory, it is too complex for instrument maintenance groups to calibrate and usually ends up being operated in manual mode. Another draw back is that both methods consist of three or more parameters within the master controller which require routine tuning to ensure system performance. This can typically exceed the technical capabilities of most instrument maintenance groups.

SUMMARY OF THE INVENTION

In view of the above, the present invention provides an improved method of controlling the temperature of a batch reactor comprising the steps of:

(a) providing a jacketed batch reactor with circulating heat-exchange fluid in the jacket surrounding the reactor and a reaction mass in the reactor, a means for heating said circulating heat-exchange fluid, a means for cooling said circulating heat-exchange fluid, a means for circulating said heat-exchange fluid to supply or withdraw heat from the reaction mass, a pair of matched temperature sensors, wherein the first temperature sensor monitors the temperature of said reaction mass in said reactor and the second temperature sensor monitors the temperature of the heat-exchange fluid in the jacket and both of said temperature sensors are operative over the same full temperature range of the reaction, a primary and slave proportional controller means arranged in a cascaded relationship, wherein said master controller is responsive to the temperature of said reaction mass by operative communication with said first

L-2

temperature sensor and wherein said slave controller is responsive to the temperature of said heat-exchange fluid by operative communication with said second temperature sensor, and (b) utilizing a cascaded master and slave controller means for controlling the temperature of said reactor wherein the temperature of the reactor monitored by said first temperature sensor serves as the measured variable, MV_1, in the master loop and the jacket temperature monitored by said second temperature sensor serves as the measured variable, MV_2, in the slave loop and the feedback for the master loop, FB_1, is set equal to said measured variable, MV_1, or the master controller set point, SP_1, thus creating a proportional variable reset.

In one embodiment of the invention, the controller circuit involves a digital electronic controller or corresponding analog electronic controller and in another embodiment a pneumatic controller is employed.

Thus, it is the primary object of the present invention to provide an improved temperature control system and associated proportional with variable bias algorithm that will allow for the heating as rapidly as physically possible during temperature ramp steps yet simultaneously allows for virtually no temperature overshoot or undershoot and will maintain a temperature control setpoint during a subsequent hold period. It is a further object to provide a temperature control system that affords the operator the ability to time the temperature control by setting only one parameter (i.e., turning only one control knob) which leads to simplicity and ease of operation. Fulfillment of these objects and the presence and fulfillment of additional objects will become apparent upon complete reading of the specification and drawing in combination with the attached claims.

BRIEF DESCRIPTION OF THE DRAWING

FIG. 1 is cross sectional schematic view of a jacketed batch reactor according to the present invention with cascaded proportional master and slave control loops with bias feedback.

FIG. 2 is a set of three segments of temperature recordings during the actual use of the improved temperature control method according to the instant invention while running a polymerization reaction in a jacketed reactor as shown in FIG. 1, wherein FIG. 2a is the initial two stage start up and temperature rise, FIG.

L-3

2b is the sustained temperature hold and FIG. 2c is the cool down.

FIG. 3 is a schematic view of a pneumatic controller alternative embodiment according to the present invention with proportional variable reset (i.e., with bias feedback) to be used in the cascaded master control loop.

DESCRIPTION OF THE PREFERRED EMBODIMENTS

The improved batch reactor control system according to the present invention, how it is made and functions and how the overall method differs from prior art as well as its advantages relative to the prior art can perhaps be best explained and understood by reference to the drawing and by reference to the under lying control algorithm associated with the use of what we have chosen to call "proportional variable reset" (PVR). As illustrated in FIG. 1, the equipment associated with the typical batch chemical reactor, generally designated by the number 10, including the temperature control system will involve reactor vessel or kettle 12 containing the reaction mass (not shown) enclosed or surrounded by a jacket 14 containing a circulating heat-exchange media or fluid 16. The reactor vessel 12 in this particularly preferred illustrated embodiment is further equipped with a stirring mechanism 18 while the heat exchange fluid 16 is withdrawn from the upper portion of the jacket 14 by circulating pump 20. Down stream from the pump 20 the circulating fluid stream is split such as to pass a portion of the heat-exchange fluid through a heater 22 and the rest of the fluid stream through a cooler 24 before recombining the streams adjusted via valves A/C and returning the fluid to the bottom of the jacket 14. It should be appreciated that many alternative variations in the specific details of the equipment including but not limited to vessels, pumps, valves, piping and the like can be employed or not employed, all as generally well known in the chemical arts without, departing from the spirit and scope of providing a jacketed batch reactor with circulating heat-exchange fluid as the phrase is used herein to describe and claim the instant invention.

As further illustrated in FIG. 1, the reactor vessel 12 is equipped with a first temperature sensor 26 which in use monitors the temperature of the reaction mass while simultaneously a second temperature sensor 28 monitors the temperature of the circulating heat-exchange fluid as it enters the lower portion of the jacket 14. It is critical for

L-4

purposes of this invention that the respective temperature sensors be selected such that they are matched. By this it is meant that the temperature range of the slave (inner) control loop and the temperature range of the master (outer) control loop be equal either through transmitter range selection, internal/external scaling or the like. As can be seen from FIG. 1, the signal from the first sensor 26, corresponding to the temperature of the reactor, is operatively connected to and transmits the temperature (TT) to the master or outer temperature controller 30 while the signal from the second sensor 28, corresponding to the temperature of the jacket, is operatively connected to and transmits the temperature (TT) to the slave or inner temperature controller 32 of a pair of cascaded temperature controllers (again 30 and 32, respectively). A transducer (I/P) transduces the signal from electronic to pneumatic as necessary. As such the process or measured variable for the master control loop is the temperature of the reactor; i.e.; MV_1(Reactor Temp). Similarly, the process or measured variable for the slave control loop is the temperature of the jacket; MV_2(Jacket Temp). Again the pair of controllers 30 and 32 are operatively connected and thus arranged in a cascade control architecture with reactor jacket temperature as the inner or slave loop and reactor temperature as the outer or master loop. The feed back signal in the master controller, which is normally equal to controller output (OUT_1) or slave loop process Variable (MV_2), is for purposes of this invention set equal to master loop process variable (MV_1) or the master controller set point (SP_1) or other parameter that tracks or parallels one of these variables. As such and for purposes of claiming this invention the reference to setting the feedback for the master loop (FB_1) equal to the measured variable (MV_1) is intended to include the reactor temperature and equivalent parameters such as the reactor setpoint or the like.

More specifically and for further clarity, in the generic PID controller algorithm the output equals a controller gain times the difference between the set point and the measured process variable plus the value of the external feed back variable as represented by:

OUTPUT=K(SP-MV)+FB

where;

K=controller gain

L-5

SP=set point on controller

MV=measured process variable on controller

FB=external feed back variable

Thus in the case of the instant cascaded pair of controllers the corresponding generic algorithm for the "Proportional with Variable Reset" is represented as:

TC_2 =Slave Loop

$OUT_2 = K_2 (SP_1 - MV_2) + FB_2$

and

TC_1 =Master Loop

$OUT_1 = K_1 (SP_1 - MV_1) + FB_1$

where

$FB_1 = MV_1$ (Reactor Temp) or $FB_1 = SP_1$ (Reactor SP)

and thus

$OUT_1 = K_1 (SP_1 - MV_1) + MV_1$ or $OUT_1 = K_1 (SP_1 - MV_1) + SP_1$

but MV_1 becomes equal to SP_1 at the control point thus at temperature hold subsequent to optimum rapid temperature rise

OUT_1 =K(zero)+MV_1 or OUT_1 =K=(zero)+SP_1

and

OUT_1(Jacket SP) =MV_1(Reactor Temp) or OUT_1 (Jacket SP)=SP_1 (Reactor SP)

In other words, in the embodiment illustrated in FIG. 1 wherein the feedback for the master loop, FB_1, is set equal to the measured variable, MV_1, the OUT_1 (which is the Jacket SP, see tie line in FIG. 1 between 30 and 32) becomes MV_1 at temperature hold subsequent to optimum rapid temperature rise and this value is the reactor temperature at that hold. Similarly for the alternate embodiment represented by the above alternate equations wherein the feedback, FB_1, is set equal to the master controller set point, SP_1, the OUT_1 at L-6

temperature hold becomes SP_1 as well as the jacket SP which again is the desired targeted reactor temperature at hold.

Again in contrast, normal batch control would set

$FB_1 = OUT_1$ or MV_2.

which would not accomplish the end result that the PVR algorithm achieves. By way of further explanation an ideal control would be to keep OUT_1 at a maximum until the reactor reaches the required setpoint and then set the jacket temperature setpoint at the kettle setpoint. However, on a pragmatic level this concept would require extraordinary effort using contemporary cascaded PID controllers. In comparison the instant invention approaches this and involves merely tuning the system by adjusting one gain setting. In the following Example there is a temperature ramp step from 200° C. to 230° C. In the ideal case the OUT_1 would be kept at a maximum of perhaps 300° C. by setting SP_2 on TC_2 at this value until the reactor reaches the required setpoint 230° C. and then one would reset OUT_1 to 230° C. as represented by:

if MV_1(Reactor temp) not=230° C.

then keep OUT_1(Jacket SP) =300° C.

and

if MV_1(Reactor temp) =230° C.

then set OUT_1(Jacket SP) =300° C.

According to the instant PVR the

$OUT_1 = K_1 (SP_1 - MV_1) + MV_1$

and at the control point

$MV_1 = SP_1$

thus

$OUT_1 = K(\text{zero}) + MV_1$

or more specifically

OUT_1(Jacket SP) $= MV_1$(Reactor Temp). L-7

and

OUT_1 =230° C.

In other words, the algorithm associated with the instant control system inherently mimics the ideal state while the only operator adjustment is the controller gain. It is well within the skill of the operator to adjust gain based on experience derived relative to the particular reaction to be rim. One merely determines empirically the gain setting necessary to achieve simultaneously optimum temperature rise and/or fall without overshoot and/or undershoot. Again a concept well within any practitioner's skills,

The actual choice and installation of the cascade controller is not viewed to be critical in that it is felt that any such device as generally practiced in the art can be employed. Preferably a general purpose PID digital cascade controller is to be employed with reassignment of the Master Loop feedback as described above. It should be further appreciated that a corresponding analog electronic controller or a pneumatic controller can be and has been successfully implemented according to the PVR algorithm. One specific embodiment of such a configuration is schematically illustrated in FIG. 3 wherein the K, MV, SP and FB have the same meaning as illustrate above and the general output, θ_{out}, is given by:

θ_{out} =K(SP-MV)+FB.

Commercially available controllers which have been successfully used in the improved temperature control system according to the present invention include Moore Products' 352 Single Loop Digital Controller, Moore Products' analog pneumatic controller and Honeywell's TDC 3000 Distributed Control System.

EXAMPLE

To further illustrate and evaluate the improved batch reactor temperature control method employing the cascaded master and slave control loops with variable bias feedback according to the present invention, a five hundred gallon commercial jacketed batch chemical reactor arranged and configured as shown in FIG. 1 and as described herein was employed to perform a polymerization reaction. This polymerization reaction L-8

produces a commercial polymer that is used as a fiber finish additive commonly applied to polyester fabrics as a slickener and commercially available from E. I. du Pont de Nemours and Company under the tradename Zelcon.RTM.. The commercial scale batch reaction is known to be troublesome in that temperature variations during reaction and lack of consistent reproducibility of temperature from batch to batch leads to variations in molecular weight distribution and viscosity. This in tem represents an inability to meet product quality specifications on a consistent and reproducible basis.

As configured, the commercial scale 500 gallon batch reactor was characterized as having a heat input capability 2.5 times the heat removal capability with a 30 minute dead time to heating or cooling changes. Thus the heating and cooling ramps took hours (see FIG. 2a, 2b and 2c). The heating was by an electric heater which had two elements and cooling and heating were split range. The control requirements for the reaction involved heating the reaction mass to a minimum of 100° C. as fast as possible followed by the manual addition of a charge of Carbowax. This took approximately 1 hour and 45 minutes. The reaction mass was then heated to 200° C. as fast as possible which took another 2 hours. A temperature hold at 200° to 205° C. for 1 hour followed the second temperature ramp after which the reaction mass was again heated with application of vacuum (see endotherm of FIG. 2a) as fast as possible to 230° C. which involved another 3 hours. At this point the reactor was held at 230°±2° C. (see FIG. 2b) until the desired viscosity set point is reached. This typically takes from 8 to 12 hours and is the most critical control step for product quality. The reactor was then cooled to 180° C. as fast as possible which took 7 hours after which the product was transferred from the reactor.

As shown in the temperature recordings of FIG. 2a, 2b and 2c, the temperature rise-times of the reactor (the lower smooth curve) were very rapid (i.e., optimal) and the temperature control during the critical hold times (in particular see FIG. 2b) were very flat and stable (i.e., again optimal). More importantly, the overall reaction process and product quality has been found to be highly reproducible particularly when compared to the previous history of such reaction. Also, implementing the process using pneumatic circuits rather than electric has been shown to give virtually identical results again indicating that the use of the proportional variable reset concept (i.e., in this case the temperature of the reactor as the L-9

feedback to the master temperature controller) significantly improves the temperature control of the batch reactor.

The advantages and benefits associated with the improved batch reactor temperature control system according to the instant invention are felt to be numerous and significant. For example, the improved temperature control system with proportional variable reset allows for the heating as rapidly as the possible during temperature ramp steps yet simultaneously allows for virtually no temperature overshoot or undershoot. In other words, the proportional with variable bias algorithm will heat and cool at the maximum heat input and removal rate of the process equipment as rapidly as process dynamics allow. It will maintain a temperature control setpoint during hold period with no over shoot or undershoot after maximum heating and cooling ramp. The system affords the operator the ability to time the temperature control by setting only one parameter (i.e., mining only one control knob) which leads to simplicity and ease of operation. Also, the product quality, ease of meeting production specifications and reproducibility is markedly improved particularly for reaction that are highly sensitive to temperature fluctuations during manufacturing. These advantages and benefits translate into economic savings in use of human resources, equipment utilization (such as reduced cycle time) and reduction of off-spec product.

Having thus described and exemplified the invention with a certain degree of particularity, it should be appreciated that the following claims are not to be so limited but are to be afforded a scope commensurate with the wording of each element of the claim and equivalents thereof.

* * * * *

Other References

Instrumentation for Process Measurement and Cotrol Norman Anderson, 1974 pp. 158, 159.

L-10

- EXHIBIT M -

Augusta Newsprint Company

2434 Doug Barnard Parkway
Augusta, GA 30906
Phone 706 798 3440
Fax 706 312 6473

Final Scientific Report

Implementation of a TMP Advanced Quality Control System at a Newsprint Manufacturing Plant

Award Number: DE-FC36-01ID14210

Time Period: 8/14/01 to 8/15/05

Mill Contact: Sébastien Kidd
706 798 3440 x413
skidd@augustanewsprint.com

Project Director: Douglas W. Hooker/DOE
303 275 4753
doug.hooker@go.doe.gov

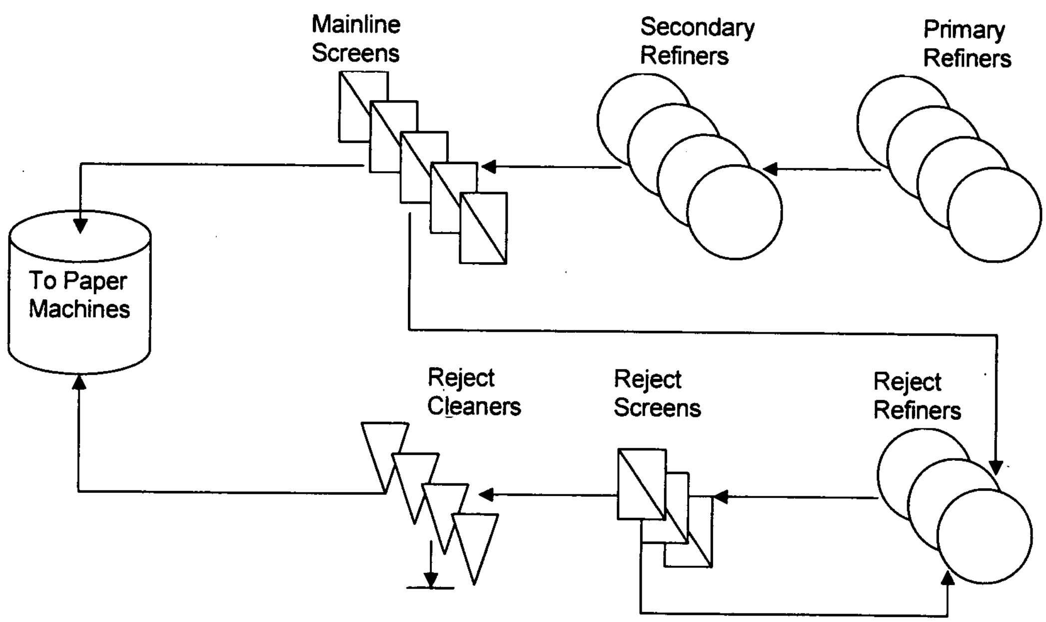
Project Team: David Roberston/DOE
Valri Robinson/DOE
Bill Strand/Pacific Simulation Metso
Sébastien Kidd/Augusta Newsprint

Executive Summary:

This project provided for the implementation of an advanced, model predictive multi-variant controller that works with the mill's existing distributed control system. The method provides real time and online predictive models and modifies control actions to maximize quality and minimize energy costs. Using software sensors, the system can predict difficult-to-measure quality and process variables and make necessary process control decisions to accurately control pulp quality while minimizing electrical usage. This method of control has allowed Augusta Newsprint Company to optimize the operation of its' Thermo Mechanical Pulp mill for lower energy consumption and lower pulp quality variance.

Augusta Newsprint's Thermo Mechanical Pulp(TMP) mill is one of two pulp mills that feed two high speed Newsprint Machines. The TMP mill is comprised of eleven 12,000 horsepower refiners. These refiners, in three stages, take wood chips from approximately 1" x 1" x ¼" to individual pliable fibers. This process is very energy intensive and can change due to the conditioning of the chips and the operation of the rest of the TMP process, primarily the rate at which fibers are rejected and re-refined.

Figure 1 -Thermo Mechanical Pulp Process Flow Diagram



Results:

The primary goal for this project was the reduction of energy consumption for the mill. Figure 2 shows a savings of ~ 3.5% for the total mill, or $1.25 million per year at 2001 energy prices. The goal of the project was to save $1.12 million per year after complete implementation; this is judged on a quarterly basis using a sliding scale of energy usage versus pulp quality (seen in Figure 3). In addition to the energy savings; a goal of 40% pulp quality variance reduction was set (progress can be seen in Figure 4). Other anticipated and recognized benefits were increased pulp quality and reduced usage of supplemental Kraft pulp. These results were achieved while increasing the production rate of the Thermo Mechanical Pulp mill; though many other factors in the mill assisted in these results, i.e. chemical additions, pulp quality targets and various energy reducing projects in the mill.

Figure 2 – Total Mill Energy Consumption

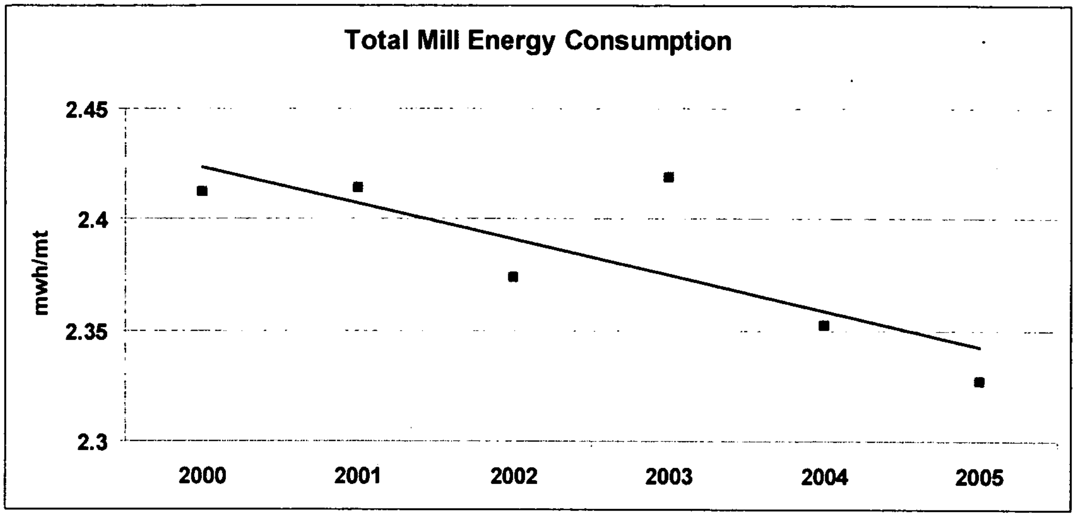


Figure 3 – Burst versus Specific Energy

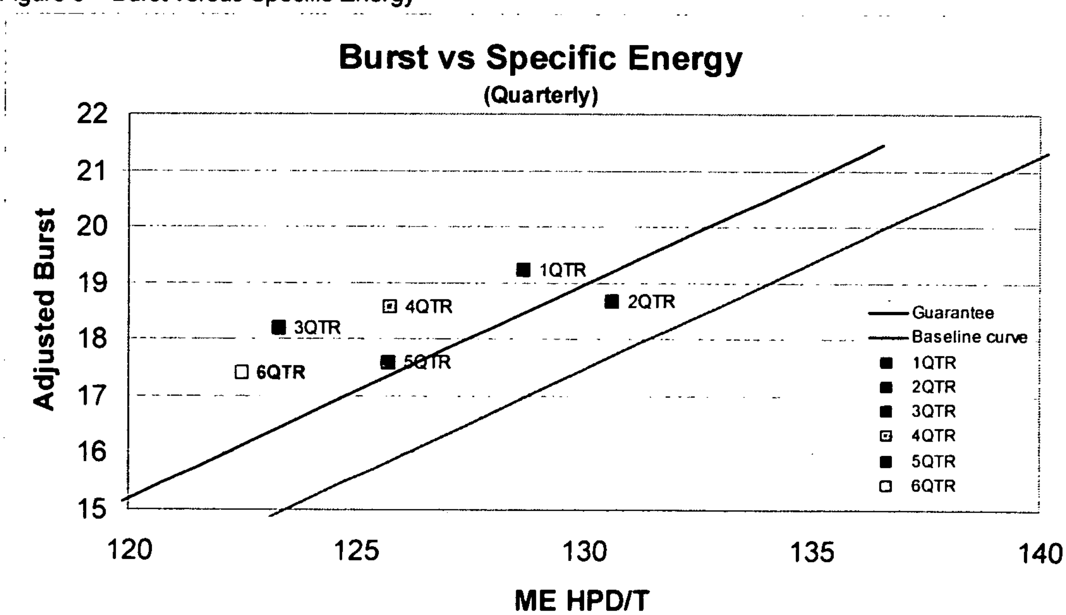
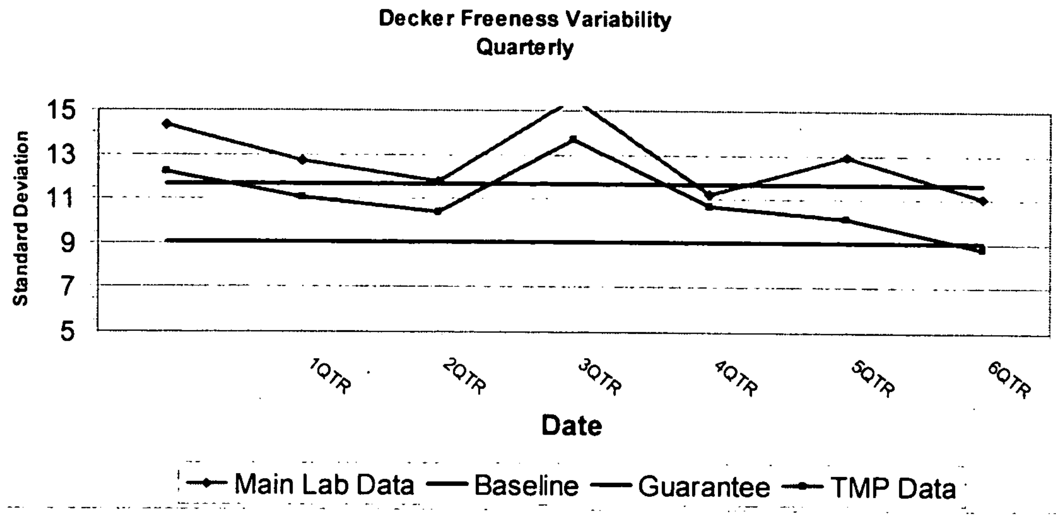


Figure 4 – Decker Freeness Variability



Project Activities:

Initial Data Analysis
Prior to beginning any mill Advanced Quality Control system, existing historical data was analyzed. The purpose of this analysis is to determine preliminary estimates in the following areas:

- What are the primary areas of control (refiners, screens, blending)?
- What is the present pulp quality and paper quality variance? How does it compare against other mills?
- What are the economic factors affecting this project:
    - Furnish (Kraft, broke, TMP, RNP)
    - Energy (electrical, steam, etc..)
    - Paper Machine Efficiency (speed, breaks, off-spec, etc...)
    - Chemical Costs (bleach, retention aid, filler, pitch control, etc...)
    - Chip Costs...

Control Infrastructure
Advanced Quality Control projects are designed to use a central process control data server. This project task involves the design and implementation of the process control data server. Normally there are two interfaces installed:

- DCS interface – This is for all continuous process data.
- MIS interface – Most off-line testing (pulp and paper quality) is stored in a MIS system. At Augusta Newsprint, PI is the primary system used for data storage.

Initial Simulation
A detailed steady state and dynamic simulation was developed of the mill operation from the chip pile through the pulp mill. The simulation, developed using WinGEMS 5.0, serves a number of purposes:

- Baseline the mill operation
- Identify process bottlenecks
- Further refine the availability of data
- Determine significant time delays in the system for synchronizing process and quality data
- Initial basis for tuning full mill control systems
- Platform for building control systems for the operators

Time Synchronization Simulation
When predicting in real time, the effect of pulp quality at various points in the pulp mill must be time determined, data from throughout the mill must be time synchronized. Time synchronization must deal with the following factors:

- Changing tank levels
- Changing production rates
- Process recycles
- Dead time, mixed tanks as well as non-ideal flow

A specialized WinGEMS simulation called an Event Profiler was developed for determining the process delays. The simulation was first built and tested off-line, then linked to the process control data server for on-line real time synchronization.

Evaluation of Regulatory Control
Advanced Quality Control requires that the regulatory control elements and control loops are designed, implemented, tuned and maintained correctly. A complete evaluation of the applicable regulatory loops was performed. Changes to the regulatory loops and elements were recommended prior to proceeding with Advanced Quality Control.

Detailed Unit Operation Modeling
Whenever possible, historical data or mechanistic models are used to develop the control parameters for the Advanced Quality Control system. However, a portion of the models required for building the controllers does not lie in the historical data analysis or in the first principle mass and energy balances. Mill trials were performed to identify these model coefficients. This was normally in the form of process bump testing.

Detailed Control
Advanced Quality Control has the goal of improving the economics of the process by achieving sustainable and quantifiable economic benefits. Sustainable economic benefits are achievable through coordination of the Advanced Quality Control from throughout the mill. Three process areas were identified for Advanced Quality Control by the initial data analysis: Mainline Refiners, Reject Refiners and the Screen Room.

Post Audit
Once all systems were complete, the system performance was audited to determine the performance level.

Issues during the Process:

- From the beginning of the project it was known that many of the existing regulatory control hardware would have to be upgraded; flow meters, consistency meters, hydraulic controls, etc…

- Separate Energy Reducing Projects; Augusta Newsprint invested an additional $450,000+ from 2002 to 2004 to reduce energy usage.

- Production Rate of the Thermo Mechanical Pulp mill; this rate which can be driven by outside market forces dramatically impacts the mill's power consumption. The higher the production rate the higher the total mill's energy consumption. 2001, 2002, 2003 and 2004 all had increases in the Thermo Mechanical Pulp mill production rates.

- Major process changes;

    - Mill chose to add a chemical treatment to the pulp that improved strength properties of the pulp,

M-5

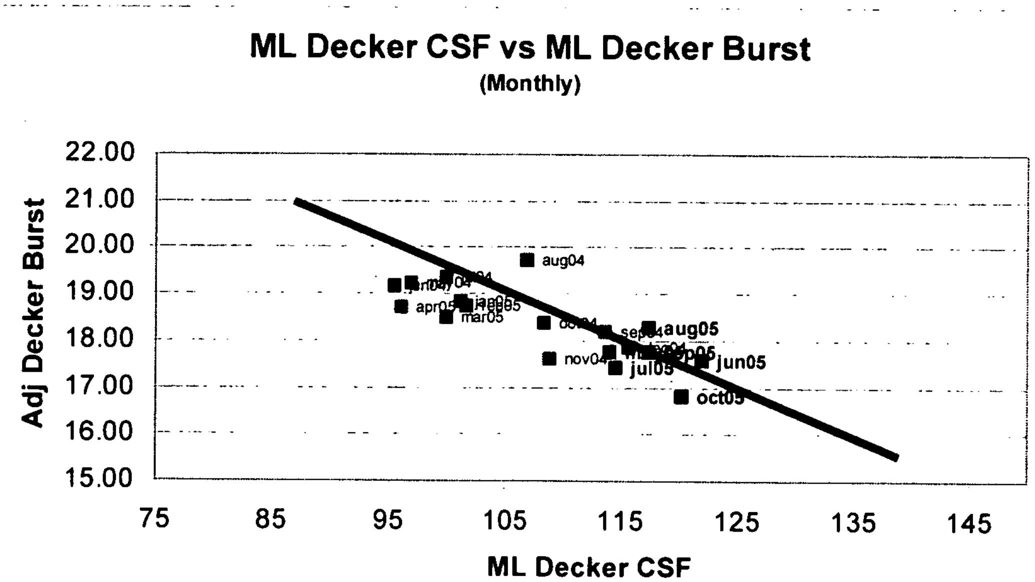
- Mill reconfigured the Mainline Screens, going from five primary screens to three primary screens and two secondary screens in a feed forward configuration.

- Operational changes in other areas of the mill; during the time period of the project, due to market conditions the paper machines needed a pulp that drained faster.

These issues were dealt with using originally twice weekly strategy meetings to now once weekly meetings. The participants for these meetings are both Pacific Simulation representatives and Augusta Newsprint representatives. Pacific Simulation provided both local and call-in representatives, while Augusta Newsprint representatives included those directly involved with the Thermo Mechanical Pulp mill operations as well as those who could speak for the other areas of the mill.

One of the major shifts in strategy has been a change in pulp freeness (a measure of how quickly water will drain from a pulp solution). This helped the machines run faster, requires less energy to achieve and helps utilize the change in strength due to chemical addition. Figure 5 shows the shift in freeness and the resulting shift in burst (a common measure of pulp strength). Due to this shift, consideration had to be taken when determining the value of energy reduction due to the project versus the energy reduction by merely shifting the freeness target. Much of the rise in the target was made possible by the lower pulp quality variance.

Figure 5 – Final Freeness versus Burst

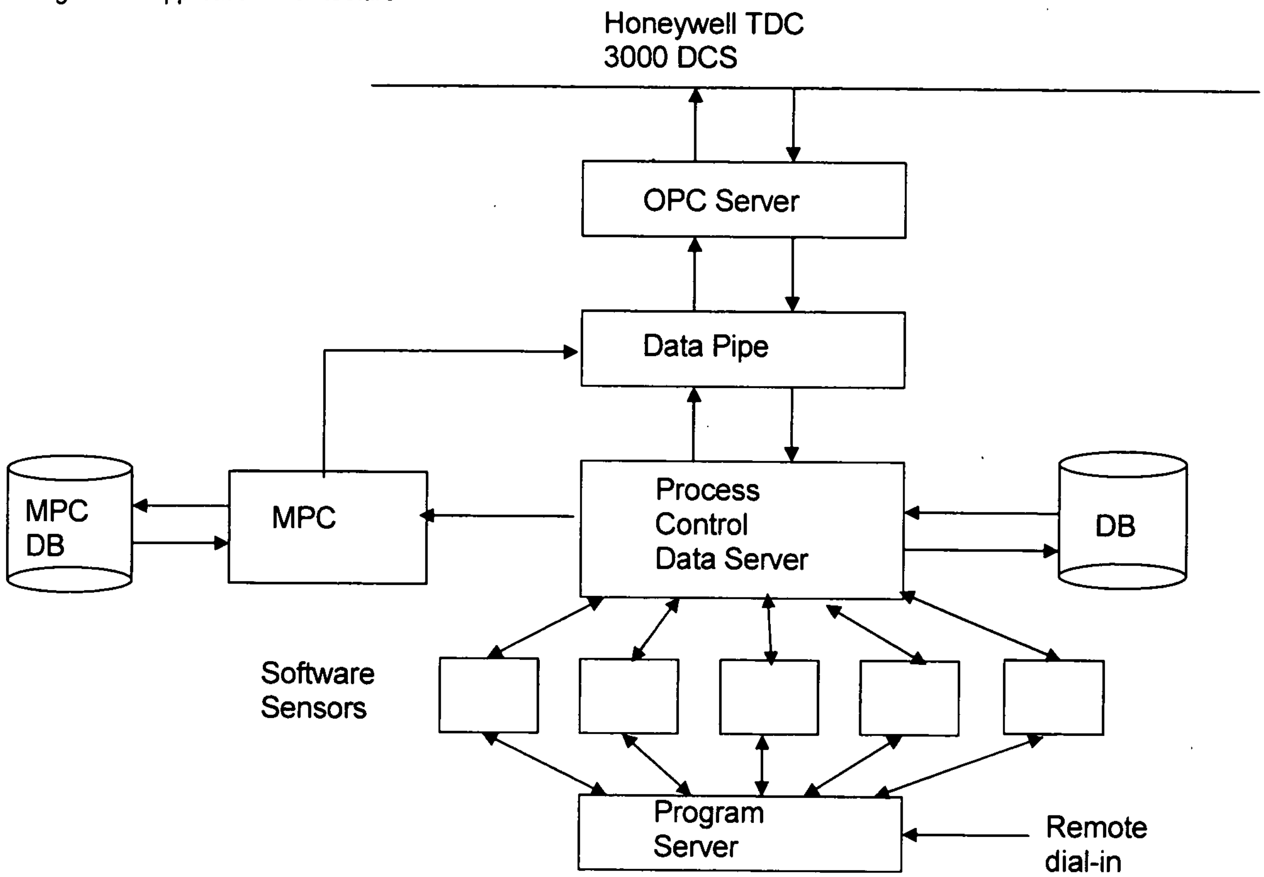


M-6

Computer Modeling:

Operating Environment
The TMP MPC (Model Predictive Controller) application runs on two operating environments, a host distributed control system interfacing directly with the process and a PC/NT server running the MPC/Process control data server application software. Figure 6 shows the basic application architecture. The arrows indicate exchange of data between the operational units (blocks).

Figure 6 – Application Architecture



Multi-Variable Predictive Controller
The MPC package is a Multi-Variable Predictive Controller, written in Visual Basic and Fortran. The optimization engine of MPC is a Fortran program called Minos, which was written and developed at Stanford U. in Connecticut.

MPC has a user-interface for setting tuning parameters and monitoring the control operation. Basic configuration is done in a flat file that specifies manipulated variables, controlled variables, their tag names, and their model-relationships. The configuration information is stored in an MS Access format database.

Data is exchanged with the DCS via MS OLE for Process Control (OPC) and the process control data server interfaces. Every 30 seconds, MPC reads process feedback (i.e. current values of the controlled and manipulated variables), performs the optimisation that calculates the new manipulated variable setpoints and sends them to the DCS.

M-7

Process Control Data Server
The process control data server provides a database and an interface to the Data Pipe. The Data Pipe transfers data between the process control data server and the OPC server. The process control data server polls data pipe for data and historizes it in the database. It is the process control data server through which the MPC application reads data from the DCS. Manipulated variable setpoints and other information are sent to the DCS via Data Pipe from IMAS.

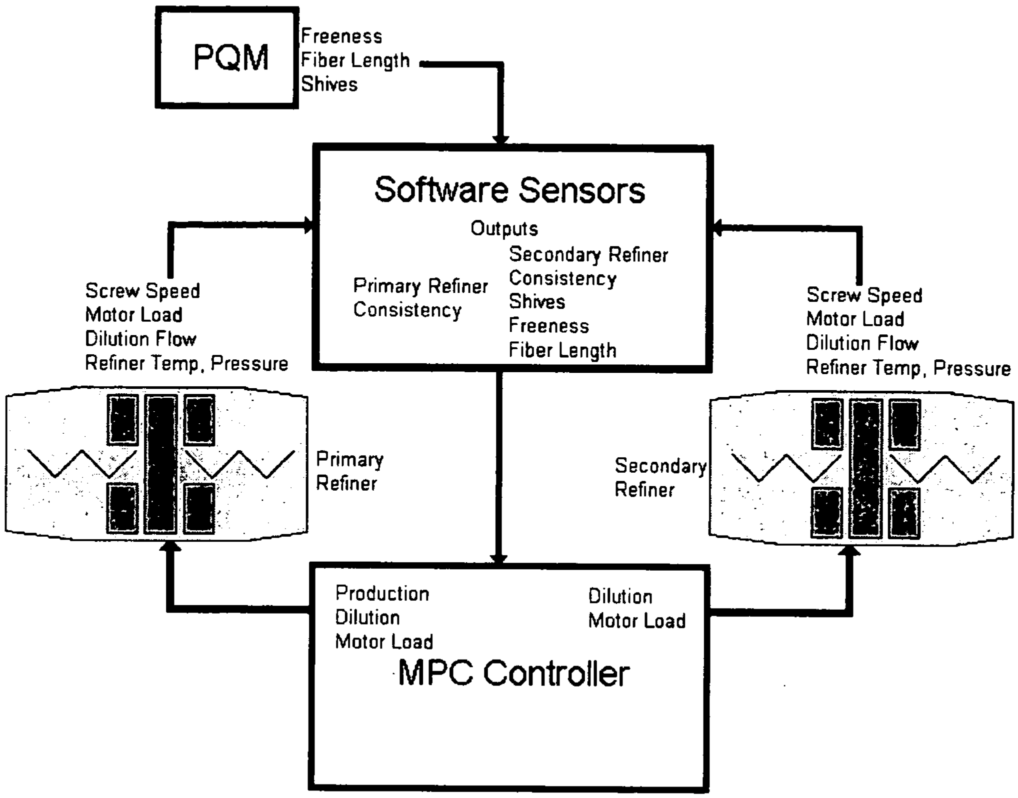
Program Server
The program server allows remote dial-in and runs Data Flow Engine applications. The Data Flow Engine applications perform scheduled calculations and interface with the process control data server. A Data Flow Engine application has been developed which calculates "Software Sensor" or model estimates of some of the TMP process variables (see Quality Variable Modeling – Software Sensors Section). These estimates are used by the MPC for controlling the TMP process.

Distributed Control System
A Honeywell TDC 3000 is the distributed control system (DCS) platform for this application. The MPC interface and the low-level PID control loops nearest to the process run on this platform. Setpoints calculated by MPC are sent to these low-level PID control loops (also known as manipulated variables).

Multi-Variable Predictive Control (MPC) Configuration
The MPC configuration includes determining the manipulated and controlled variables, defining the interaction matrix, building the input-output models and defining the optimisation objectives (e.g. maximise production).

Control System Structure and Operation
The control system structure, shown in Figure 7 includes the MPC controller and the Software Sensors. The MPC controller will take controlled variables (process outputs) from the Software Sensors and calculate control moves for the manipulated variables (process inputs).

Figure 7



Manipulated Variables (Process Inputs)
Manipulated variables (process inputs) are the low-level PID controllers used in MPC to move the process to its desired operating state. Based on a regulatory audit, the key low-level loops that influenced pulp quality were evaluated. The manipulated variables used in this application are transfer screw speed, primary motor load, secondary motor load, primary dilution flow and secondary dilution flow for the main line refiners. The manipulated variables used in the reject refiner portion of this application will be motor load, refiner dilution flow, and press feed flow. The manipulated variables for the primary screens will be the five main screen reject flows. The manipulated variables for the reject screens will be the five reject screen reject flows.

Outputs from MPC have to be validated against high and low limits, before being accepted as remote setpoints for the manipulated variables. The MPC application will not allow the manipulated variable setpoints to stay outside these "hard" high and low limits. If a manipulated variable is outside a limit, the MPC application will move the setpoint towards the limit at the maximum allowable rate defined by the controller (see MPC Tuning Section).

The refiner motor load is a key manipulated variable and is controlled by changing the plate position or gap through changing the hydraulic pressure. There is a potential risk here of pinching or plate clashing if the gap gets too narrow. A sure sign of this event happening is when the motor load/plate gap gain inverts (i.e. motor load starts to decrease with increasing plate position). A motor load control strategy has been developed that monitors this gain and automatically backs off the plate position.

Controlled Variables (Process Outputs)
Controlled variables (process outputs) are the measured process and pulp quality variables, which cannot be changed directly by MPC, but are correlated to the manipulated variables (process inputs). They are selected based on available measurements and mill quality criteria. The key controlled

M-9

variables are freeness, fiber length, shives, refiner blow line consistency, primary to secondary motor load ratio, average transfer screw speed, total specific energy, and total mill power demand. In addition, the key controlled variables for the reject refiner portion of this application are freeness, fiber length, shives, reject refiner blow line consistency, average reject freeness, average reject fiber length, average long shives, and unrefined rejects chest level. The key controlled variables for the primary screens are accepts freeness, accepts fiber length, accepts shives, and mass reject rate. The key controlled variables for the reject screens are accepts freeness, accepts fiber length, accepts shives, and mass reject rate. Each is available as a Software Sensor value from a data engine located on the NT server. The values are updated every 30 seconds.

"Synthetic" Controlled Variables
Synthetic controlled variables are those derived from the basic process variables. The ones to be used in the MPC application are:

1. TMP average screw speed, i.e. the average of all the individual line transfer screw speeds.
2. Motor load ratio, i.e. the ratio of primary refiner motor load to the total for the line.
3. Main-line long shive, i.e. the production weighted average of the individual line long shive.
4. Main-line freeness, i.e. the production weighted average of the individual line freenesses.
5. Main line fiber length, i.e. the production weighted average of the individual line fiber lengths.
6. Reject average long shive, i.e. the production weighted average of the individual line long shive.
7. Reject average freeness, i.e. the production weighted average of the individual line freenesses.
8. Reject average fiber length, i.e. the production weighted average of the individual line fiber lengths.

Quality Variable Modelling – Software Sensors
MPC requires a stable quality variable measurement that is updated frequently enough to be able to reflect process dynamics, normally found through response testing. While some of the key measurements are available on the PQM, the update interval is only 30 minutes. More frequent measurements are therefore required for continuous control. Other process variables, such as refiner blow line consistency have no on-line measurements and therefore require a continuous estimate for control.

TMP pulp quality models were implemented on the data-flow engine platform located on the NT server. The outputs from the models are called on-line estimators or software sensors. There are software sensors for long shive, fiber length, freeness and blow line for each main line and reject line and for the pulp leaving the main line latency chest and the screens. The basis for modeling long shives, fiber length, and freeness is comminution modeling. The basic inputs for the model relationships are indicated below:

Shives Out = f (Shives In, Specific Energy)
Long Fiber Out = g (Long Fiber In, Shives In, Specific Energy)
Freeness Out = h (Freeness In, Specific Energy)

The above software sensors require inputs such as production rate (transfer screw speed), dilution flows, refiner motor loads, inlet chip moisture, and refiner temperature and pressure. All are available in the DCS, except for chip moisture, temperature and pressure measurements, which can be estimated with some accuracy.

Some of the software sensors are corrected for model error – that is the difference between the PQM measurement and the model output. Every time the PQM outputs a new measurement, the model error is calculated, filtered in time and applied as a correction to the quality estimates for each line.

Samples are taken every 4 hours from the outlet of the secondary refiners and reject refiners. The samples are tested in the lab for freeness and consistency. Daily, the lab conducts a fiber length test on composite samples from the primary, secondary and reject refiners. A hand sheet burst test from 24 hour composite samples from the secondary and reject refiners is performed daily. The results are used to calculate model error for the individual lines and to correct the software sensors for that line.

MPC Execution
The control interval for MPC is 30 seconds. Every 30 seconds MPC polls the control server database for new data from the DCS. The process feedback values for the manipulated and controlled variables are then read in and the internal values in MPC are updated. MPC then defines the control problem and runs the Minos optimization program, which calculates new setpoints for the manipulated variables.

MPC then writes an ".inp" file, which contains the new setpoints for the manipulated and variables and the steady-state estimates for both the manipulated and controlled variables. The data-pipe program reads the ".inp" file and sends the values to the DCS through the OPC interface.

MPC Control Modes
For each line, MPC has, "Predict" and "Control" mode switch, as well as an MPC "on/off" switch. These are known, respectively, as the Mode Switch and the Line Switch. "Predict" mode means MPC is predicting property values on the horizon, but not making any control actions. "Control" mode means MPC is making control actions for any manipulated variables in "MPC" mode. "MPC" mode for a manipulated variable means the low-level controller is taking remote setpoints from MPC (otherwise it's in local/auto or local/manual mode).

MPC Watchdog Timer
A watchdog timer resides in the DCS, which is used for detecting "I'm Alive" status of MPC on the NT station. It counts down from 5 minutes and is reset by MPC every 30 seconds. If MPC fails to reset it after 5 minutes the MPC Master Switch is set to "off" and the operators cannot turn on MPC until it resets the counter. A watchdog timer failure indicates that either the MPC program on the NT server has been turned off or the communication between the NT server and the DCS has failed.

Interaction Matrix
Figures 8-11 are the interaction matrixes that show the relationship between the process inputs (manipulated variables) and the process outputs (controlled variables). A '+'/'-' entry in the matrix denotes the process output is positively/negatively correlated to the process input. No entry means no correlation. The matrix was developed by process analysis and inspection of the relationships found in the Software Sensor models.

Figure 8 – Main Line Refiner Interaction Matrix

		MAIN LINE REFINER INPUTS				
		Prim. Refiner Motor Load	Prim. Refiner Dilutions	Second. Refiner Motor Load	Second. Refiner Dilutions	Transfer Screw Speed
O	Secondary Freeness	-		-		+
U	Secondary Fiber Length	-	-	-	-	+
T	Secondary Long Shive	-	+	-	+	+
P	Primary Consistency	+	-			+
U	Secondary Consistency	+	-	+	-	+
T	Motor Load Ratio	+		-		
S	Mainline Freeness	-		-		+
	Mainline Fibre Length	-	-	-	-	+
	Mainline Long Shive	-	+	-	+	+
	Specific Energy	+		+		-
	TMP Avg. Screw Speed					+
	Total Mill Power Demand	+		+		

M-11

Figure 9 – Reject Refiner Interaction Matrix

		REJECT REFINER INPUTS		
		Refiner Motor Load	Refiner Dilution	Press Feed Flow
OUTPUTS	Freeness	-		+
	Fiber Length	-		+
	Long Shive	-		+
	Consistency	+	-	+
	Unrefined Reject Chest Lvl			-
	Avg Reject Freeness	-		+
	Avg Reject Fiber Length	-		+
	Avg Reject Long Shive	-		+

Figure 10 – Primary Screen Interaction Matrix

		MAINLINE SCREEN INPUTS				
		Reject Flow No. 1	Reject Flow No. 2	Reject Flow No. 3	Reject Flow No. 4	Reject Flow No. 5
OUTPUTS	Accepts Freeness	-	-	-	-	-
	Accepts Fiber Length	-	-	-	-	-
	Accepts Long Shive	-	-	-	-	-
	Mass Reject Rate	+	+	+	+	+

Figure 11 – Reject Screen Interaction Matrix

		REJECT SCREEN INPUTS				
		Reject Flow No. 1	Reject Flow No. 2	Reject Flow No. 3	Reject Flow No. 4	Reject Flow No. 5
OUTPUTS	Accepts Freeness	-	-	-	-	-
	Accepts Fiber Length	-	-	-	-	-
	Accepts Long Shive	-	-	-	-	-
	Mass Reject Rate	+	+	+	+	+

MPC Response Models

For each matrix entry, a dynamic response model is required. This is a model of the response of a controlled variable to a step or impulse change in a given manipulated variable. For the MPC controller used in this application the model is in the form of a step response weights (as opposed to a parametric linear transform function).

As discussed in Quality Variable Modeling – Software Sensor Section, a continuous measurement of some of the controlled variables is not available at a high enough frequency that would allow direct determination of the response models (i.e. through on-line bump tests). Step response models are determined by simulation using the Software Sensor models and the dynamics of the manipulated variable process variables in response to a change in setpoint.

MPC Objectives
The main objectives of the main line MPC are as follows:

1. Maintain freeness for the main line and lines 1, 2, 3, and 4 within a range
2. Maintain fiber length for the main line and lines 1, 2, 3, and 4, above a low limit.
3. Maintain long shives for the main line and lines 1, 2, 3, and 4 below a high limit.
4. Maintain a target average transfer screw speed/production rate.
5. Maintain motor load ratio within a range (between 50 - 60%).
6. Keep total Mill power demand below a maximum limit.

The main objectives of the Reject Line & Screen MPC are as follows:

1. Maintain freeness for the reject system and freeness for lines RJ1, RJ2, and RJ3 within a range
2. Maintain fiber length for the reject system and fiber length for lines RJ1, RJ2, and RJ3 above a low limit.
3. Maintain long shives for the reject system and long shives for lines RJ1, RJ2, and RJ3 below a high limit.
4. Maintain reject refiner blow consistency for lines RJ1, RJ2, and RJ3 within a range.
5. Maintain primary screen accept freeness within a range.
6. Maintain primary screen accept fiber length above a low limit.
7. Maintain primary screen accept long shives below a high limit.
8. Maintain reject screen accept freeness within a range.
9. Maintain reject screen accept fiber length above a low limit.
10. Maintain reject screen accept long shives below a high limit.
11. Maintain unrefined rejects chest level within a range.
12. Maintain Thune press amperage loads within a range.
13. Maintain primary and reject screen Mass Reject Rate within a range.

MPC Tuning
The MPC application moves the setpoints of the process inputs (manipulated variables) to maintain the process outputs (controlled variables) within their limits. The speed or size of change that MPC will make to drive a controlled variable towards its operating limits depends on:

1. How far the controlled variable is away from its configured range.
2. Magnitude of the limit weight.
3. Existing limits on the manipulated variables

Naturally, the farther the controlled variable is from its range, the harder MPC will try to bring it back in. The larger the values of the weights on the limits, the bigger the control actions will be and the fewer steps it will take. The controlled variable limit weights also express the relative importance of getting that variable into its range. The higher the relative value of a limit weight the more attention MPC will pay to getting the variable into the desired operating range.

Limits on manipulated variables are high/low range limits and control action or move size limits. If a controlled variable is outside its range, but all manipulated variables have run up against their limits, then MPC cannot do anything further to rectify the situation. Move suppression parameters are also applied to the manipulated variables. These parameters determine how fast the manipulated variables will change to bring the controlled variables into range.

MPC tuning is a somewhat subjective exercise. A simplified list of the steps required to tune the MPC application is shown as follows:

Step 1: Prioritise the importance of keeping controlled variables within their desired limits. Assign weights to the limits according to the magnitude of the process values.

Step 2: Simulate how the controller will behave under different scenarios. Adjust the weights until the desired behaviour is observed.

Step 3: Prioritise the manipulated variables in order of the desired speed of response. The user may want some variables to respond faster than others. Apply move suppressions to the manipulated variables according to their magnitude. The higher the move suppression value the slower the manipulated variable will respond.

Step 4: Simulate how the controller will behave under different scenarios. Adjust the move suppressions until the desired behaviour is observed.

-EXHIBIT N =

PID Controllers in BESTune

The following PID controllers can be optimized by using the excellent PID auto-tuning software BESTune (see http://bestune.50megs.com for details). Theoretically, BESTune is able to optimize any PID controllers, as long as the PID equations implemented in them are known. In order to include more PID controllers in BESTune, I am asking you to give me more information about other well-known brands of industrial PID controllers (brand names, PID equations implemented, units of the three PID constants, etc). Your help will be very much appreciated.

The actual PID equations that are implemented inside these PLCs are all in discrete time or digital form. One example of the discrete time form of "Allen Bradley Logix5550 Independent PID" can be found on the web page http://bestune.50megs.com/typeABC.htm.

Variable Definition

- CO = Controller Output, also called control variable, manipulated variable, etc.
- PV = Process Variable, also called controlled variable, measured variable, etc.
- SP = Set Point, also called desired value, reference signal, command, etc.
- e = SP-PV

Allen Bradley Logix5550 Independent PID

$$CO = K_p e + K_i \int e dt + K_d \frac{d(-PV)}{dt}$$

N-1

where

Kp:	Proportional gain	No unit
Ki:	Integral gain	(1/second)
Kd:	Derivative gain	(seconds)

Allen Bradley Logix5550 Dependent PID

$$CO = K_c\left(e + \frac{1}{T_i}\int edt + T_d\frac{d(-PV)}{dt}\right)$$

where

Kc:	Proportional gain	No unit
Ti:	Reset time	(min/rep)
Td:	Rate time	(min)

Allen Bradley PLC5 Independent PID - Using Integer Blocks

$$CO = K_p e + K_i\int edt + K_d\frac{d(-PV)}{dt}$$

where

Kp:	Proportional gain	(0.01)
Ki:	Integral gain	(0.001/ second)
Kd:	Derivative gain	(seconds)

Allen Bradley PLC5 Independent PID - Using PD Blocks

$$CO = K_p e + K_i\int edt + K_d\frac{d(-PV)}{dt}$$

where

Kp:	Proportional gain	No unit
Ki:	Integral gain	(1/ second)
Kd:	Derivative gain	(seconds)

Allen Bradley PLC5 ISA PID - Using Integer Blocks

$$CO = K_c\left(e + \frac{1}{T_i}\int edt + T_d\frac{d(-PV)}{dt}\right)$$ N-2

where

Kc:	Proportional gain	(0.01)
Ti:	Reset time	(0.01min/rep)
Td:	Rate time	(0.01min)

Allen Bradley PLC5 ISA PID - Using PD Blocks

$$CO = K_c\left(e + \frac{1}{T_i}\int edt + T_d\frac{d(-PV)}{dt}\right)$$

where

Kc:	Proportional gain	No unit
Ti:	Reset time	(min/rep)
Td:	Rate time	(min)

Allen Bradley SLC5/02,SLC5/03 and SLC5/04 ISA PID

$$CO = K_c\left(e + \frac{1}{T_i}\int edt + T_d\frac{d(-PV)}{dt}\right)$$

where

Kc:	Proportional gain	(0.1)
Ti:	Reset time	(0.1min/rep)
Td:	Rate time	(0.01min)

Bailey Function Code FC19 with K=1

$$CO = K\left(K_p e + \frac{K_i}{60}\int edt + 60K_d\frac{de}{dt}\right)$$

where

K:	Gain multiplier	No unit
Kp:	Proportional gain	No unit
Ki:	Integral reset	1/min
Kd:	Derivative rate action	Min

Bailey Function Code FC156 Independent Form with K=1

$$CO = K\left(K_p e + \frac{K_i}{60}\int edt + 60K_d\frac{de}{dt}\right)$$ N-3

where

K:	Gain multiplier	No unit
Kp:	Proportional gain	No unit
Ki:	Integral reset	Resets/min
Kd:	Derivative rate action	Min

Concept PID1 - PID Controller

$$CO = Gain\left(e + \frac{1}{TI}\int edt + TD\frac{de}{dt}\right)$$

where

Gain:	Proportional gain	No unit
TI:	Reset time	(milliseconds)
TD:	Derivative Action time	(milliseconds)

Concept PID1P - PID Controller with parallel structure

$$CO = KPe + KI\int edt + KD\frac{de}{dt}$$

where

KP:	Proportional gain	No unit
KI:	Integral rate	(1/milliseconds)
KD:	Differentiation rate	(milliseconds)

Fischer & Porter DCU 3200 CON Ideal with KP = 1

$$CO = KC\left(KPe + \frac{1}{TR}\int edt + TD\frac{de}{dt}\right)$$

If Kp = 1, the above equation reduces to:

$$CO = KC\left(e + \frac{1}{TR}\int edt + TD\frac{de}{dt}\right)$$ N-4

where

KC:	Gain constant	No unit
TR:	Reset time	(min/rep)
TD:	Derivative term	(min)

Fischer & Porter DCU 3200 CON Parallel KP variable with KC=1

$$CO = KC\left(KPe + \frac{1}{TR}\int edt + TD\frac{de}{dt}\right)$$

If KC=1, the above equation reduces to:

$$CO = KPe + \frac{1}{TR}\int edt + TD\frac{de}{dt}$$

where

KP:	Proportional gain	No unit
TR:	Reset time	(min/rep)
TD:	Derivative term	(min)

GE Fanuc Series 90-30 and 90-70 Independent Form PID

$$CO = K_p e + K_i\int edt + K_d\frac{de}{dt}$$

where

Kp:	Proportional gain	(0.01)
Ki:	Reset time	(0.001rep/second)
Kd:	Derivative gain	(0.01 seconds)

Hartmann & Braun Freelance 2000 PID

$$CO = CP\left(e + \frac{1}{TR}\int edt + TD\frac{de}{dt}\right)$$

where

CP:	Proportional correction value	No unit
TR:	Reset time	(milliseconds)
TD:	Rate time	(milliseconds)

N-5

Honeywell TDC 3000 APM Non - Interactive PID

$$CO = K\left(e + \frac{1}{T1}\int edt + T2\frac{de}{dt}\right)$$

where

K:	Gain	No unit
T1:	Integral time constant	(min/rep)
T2:	Derivative time constant	(min)

Modicon 984 PLC PID2 Equation

$$CO = \frac{100}{PB}\left(e + K2\int edt + K3\frac{de}{dt}\right)$$

where

PB:	Proportional band	No unit
K2:	Integral mode gain constant	(0.01min/rep)
K3:	Derivative mode gain constant	(0.01min)

Siemens S7 PB41 CONT_C PID

$$CO = Gain * e + \frac{1}{TI}\int edt + TD\frac{de}{dt}$$

where

Gain:	Proportional gain	No unit
TI:	Reset time	(seconds)
TD:	Derivative time	(seconds)

Yokogawa Field Control Station (FCS) PID

$$CO = \frac{100}{PB}\left(e + \frac{1}{T_i}\int edt + T_d\frac{de}{dt}\right)$$ N-6

where

PB:	Proportional band	No unit
Ti:	Integral time	(seconds)
Td:	Derivative time	(seconds)

N-7

EXHIBIT O =

PID Tuning Software

January 11, 2000

QuickTune from **ControlSoft Inc.** is designed to tune PID loops for the Honeywell TDC 3000. The software uses the same advanced process control PID tuning technology that is licensed to many of the major PLC and DCS companies. A two-week free trial is available.

January 2000, RS# 201

O-1

> Contact Us

> Media Kit

> Reprints

> **Special Collections**

> Drying Files

> Energy Notes

> Heating Highlights

> HelpLine Q and A

> Safety Zone

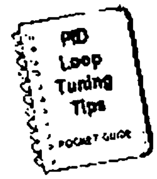
> **Related Publications**

> Process Cooling & Equipment

0-2

= EXHIBIT D =

CONTROLSOFT
ADVANCED PROCESS CONTROL MADE EASY



Request your free copy.

Company Applications Industries Products Services Support Training

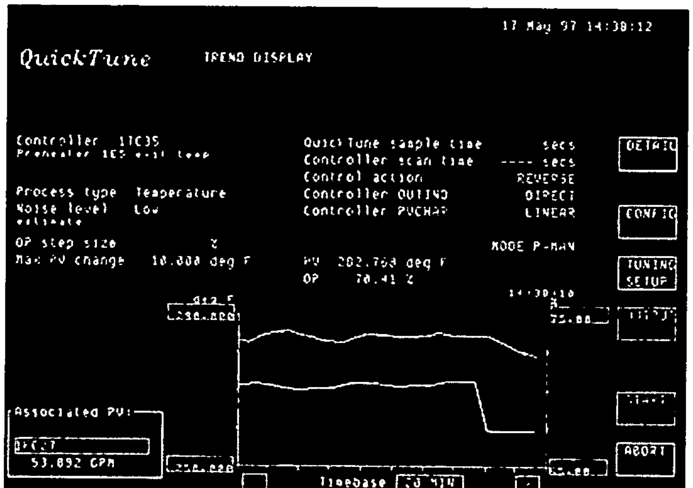
INTUNE™

Customized Solution for Users of Honeywell

QuickTune™

for PID tuning on the Honeywell TDC 3000 system

See complete details about INTUNE.



QuickTune scores high for ease of use, reliability, and ease of integration on existing Honeywell TDC 3000 systems.

Developed by ControlSoft, Inc. (U.S.A.), and Sandiford Computer Service (U.K.), QuickTune automatically generates optimal K, T1, and T2 values for PID loops.

- simple to install on the AM module
- ready to use, instantly on-line
- no connecting cables, no off-line data collection
- fast process identification
- standard TDC 3000 operator interface

D-1

Complete QuickTune user manual with every purchase!

QuickTune (version 2.01) is now available on Zip and Bernoulli disks.

For system requirements, pricing, and ordering, contact ControlSoft, Inc.

See complete details about INTUNE.

CONTROLSOFT
440-443-3900
info@controlsoftinc.com

P-2